目 录

第一章 概　论

当前电子技术已经全面地进入了数字化的时代，其发展速度是空前的。为了大幅度地提高效率，在研制新型电子系统的过程中，往往采用如下的程序：首先提出一个新的设想；然后对其进行仿真以验证该设想的可行性，并预测其性能参数；在达到了预期的效果之后，再进行硬件的实现。这种方法已经逐步成为了科研工作的一种主要模式，其中进行系统仿真是极其重要的一环。

MATLAB 语言本身就是一种对线性系统进行分析和仿真的方便工具，它特别适用于对电子系统进行计算机仿真，由于 MATLAB 的种种特点，美国和欧洲的若干著名大学在 1990 年前后就把 MATLAB 语言正式列入了电子工程专业研究生和本科生的教学计划，在工业界内 MATLAB 也已经成为工程师们应该掌握的一种工具。在我国的各高等学校内，近年来也开始逐步在研究生和本科生的教学环节中引入 MATLAB 语言，这对于改善教学效果、提高学生的科研能力无疑是极为有益的。

1.1 电子系统仿真

一般来说，一个电子系统可以抽象为由线性器件和非线性器件组成的数学模型，而电子系统仿真就是根据适当的模型对实际的电子系统进行实验研究的过程。数学模型的建立是进行系统仿真的基础，也是进行系统仿真必须首先解决的问题，数学模型的正确与否、数学模型与实际电子系统的近似程度都会直接影响仿真的结果，数学模型的建立通常又称为系统建模，它是对电子系统进行仿真的一个重要的环节。在确定了电子系统的数学模型之后，就可以采用适当的仿真语言或仿真工具对系统进行仿真了。

在电子系统的数学模型建立起来之后，还必须找到一种或几种合适的仿真算法，然后才能编制仿真程序，进而进行电子系统的仿真实验。仿真算法、仿真语言和仿真程序构成了数字仿真软件。对于电子系统来说，可以采用的仿真语言有很多种，其中 MATLAB 语言具有其非常突出的优点，故目前已经成为电子系统首选的仿真语言，在科研和教学领域内广为使用。

一般来说，电子系统的仿真过程可以分为如下的五个步骤：

(1) 根据要分析的电子系统，建立相应的数学模型

(2) 找到合适的仿真算法

(3) 应用仿真语言编制计算程序

(4) 根据初步的仿真结果对该数学模型进行验证

(5) 进行系统仿真，并认真地分析仿真的结果

上述的五个步骤之间是有连带关系的，不可能将它们完全分离开。在实际的仿真工作中，往往是反复地重复以上的前四个步骤，以保证数学模型的正确性、仿真算法的可行性、

仿真程序的准确性和可靠性，最后编制成一个成熟的仿真软件。使用此仿真软件对所研究的电子系统进行仿真，以获得大量的特性参数，从而达到使用仿真来模拟实验的目的。

常用的电子系统的数学模型和仿真算法有微分方程、差分方程、积分方程、积分变换等等，同时又可以分为时域和频域两大类；对于同一问题，往往可以采用几种方法，这要根据具体情况而定。

例如图 1.1 中所示的一阶 RC 网络，既可以采用一阶常微分方程作为它的数学模型，然后通过数值法求解此方程，进而得到它在时域内的零输入响应和零状态响应；也可以采用拉氏变换的方法，先求出其传递函数，进一步便可得到其幅频响应和相频响应。

在分析非线性电子系统时，频谱分析也是一种经常使用的方法，由于在多种仿真语言中都具备了快速傅立叶变换（FFT）功能，因此在电子系统仿真的过程中进行频谱分析是十分方便的。

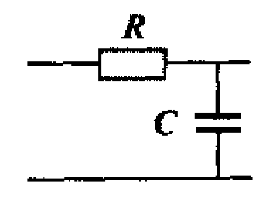


图 1.1　一阶 RC 网络

仿真技术是近年来普遍开始采用的一种研究方法，它具有容易实现、模拟实验周期短、便于对系统设计进行修改、便于对系统参数进行最优化设计、投资少、节约能源、安全性高、便于掌握等一系列突出的优点，无论是在电子系统设计、科研工作、计算机辅助教学等方面，系统仿真都得到了广泛的使用。

目前在数字通信系统的设计工作中就普遍地采用了电子系统仿真的方法。例如，海外的华裔学者就曾经使用电子系统仿真的方法对整个数字卫星广播的信道编解码环节进行了模拟，其中包括了 RS 编解码、数字调制解调器等等，其结果是令人满意的，大大缩短了整个系统的设计周期，且所需的费用较低。

另外，近年来计算机仿真的方法在国内各个高等院校电子信息专业的教学实践中应用得越来越普遍，这对于改进教学效果、给学生提供形象化的信息、提高学生的学习兴趣、提高学生的自学能力、加强学生对授课内容的理解、减少课堂教学时数、减轻教师的负担无疑都是十分有益的。电子系统仿真的方法特别适合于电子信息类本科生专业基础课程和专业课程的机辅教学工作。学生掌握了系统仿真的方法以后，不但使他加强了对所学课程的理解，同时还便于钻研一些他本人感兴趣的问题，因为在电子系统仿真的过程中，不需要购买大量的电子器件，只需有一台计算机就可以模拟实际的电子系统。显然，这样做在整体上有利于对学生分析问题的能力和解决问题的能力的培养。

1.2　MATLAB 语言的概况

MATLAB 为 Matrix Laboratory 的缩写，其本意为矩阵实验室。1980 年前后，美国学者克利夫·莫勒（Cleve Moler）在美国新墨西哥大学讲授线性代数时，编制了一种特别适合进行线性代数运算的程序语言，并称之为 MATLAB。后来，莫勒等人成立了一个名为 MathWorks 的软件公司，专门从事 MATLAB 语言的开发。目前该公司不但不断地更新 MATLAB 的版本，同时还出版一本名为 MATLAB News & Notes 的季刊和名为 MATLAB 文摘的电子通报月刊。该公司的网址为 www.mathworks.com，访问该站点可以获得有关 MATLAB 的最新和最权威的消息，图 1.2 为 MathWorks 公司的主页。

MATLAB 最早用于进行线性代数的辅助教学，它是以复数矩阵为基本单元的一种程序语言，提供了各种繁杂的线性代数的运算功能，并具有较方便和完善的绘图功能，故受到使

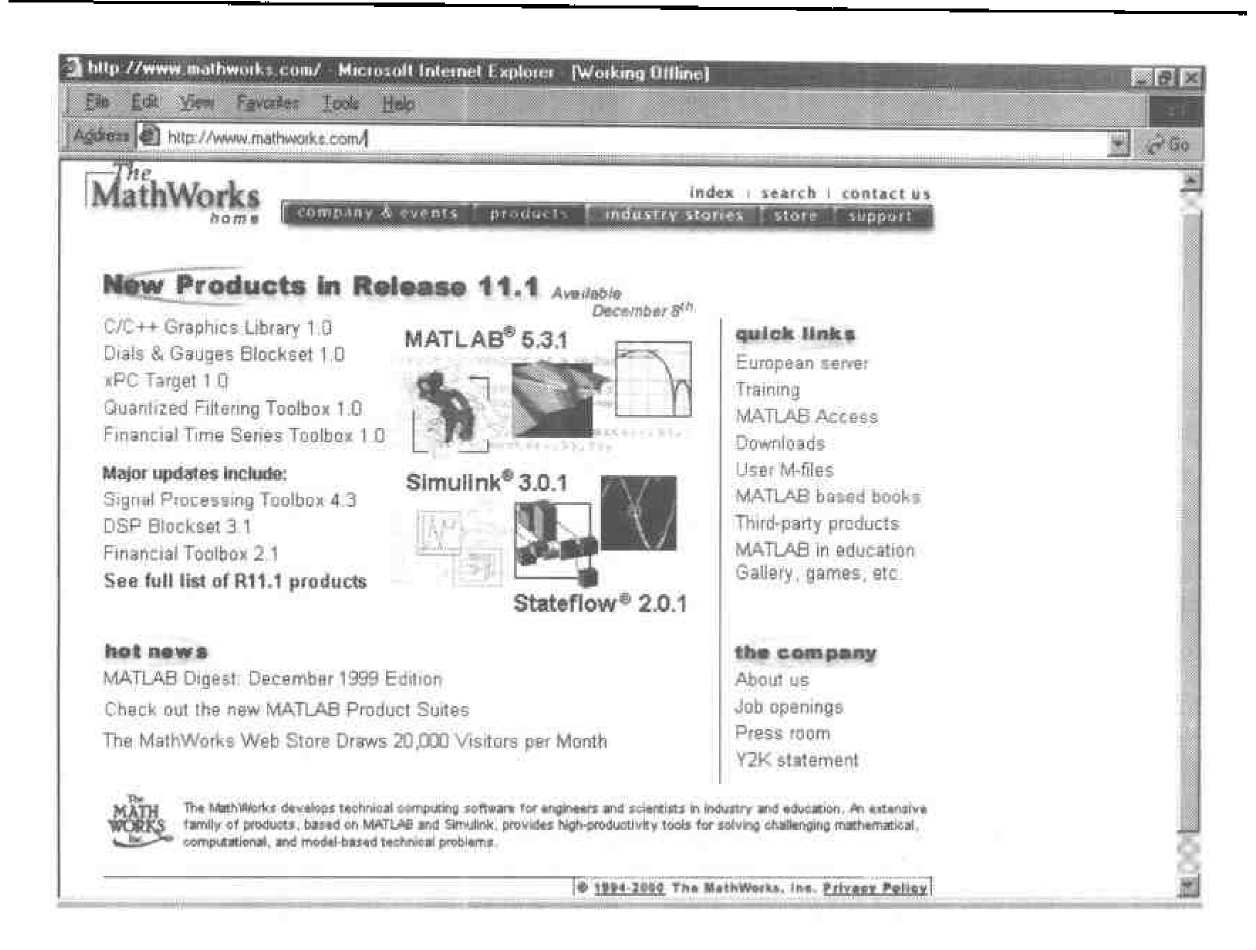


图 1.2 MathWorks 公司的主页

用者的青睐。早期的 MATLAB 语言是在 DOS 环境下运行的，由于该语言的功能很强，同时具有可扩充性，使用它可以很方便地编写出各种专用和通用的程序来，因此国外的一些著名学者在它的基础之上编写出了若干 MATLAB 的专用工具箱，这些工具箱可在诸如自动控制、通信与电子、统计等领域内使用。

目前，有人称 MATLAB 是一种全面的工程计算语言，是一种功能十分强大的工程计算及数值分析软件，它的应用范围涵盖了科学研究、高等教育、工业、电子、医疗、建筑等领域。

MATLAB 语言的主要特点是：

1. 具有丰富的数学功能，其中包括矩阵各种运算，如正交变换、三角分解、特征值、常见的特殊矩阵；包括各种特殊函数，如贝塞尔函数、勒让德函数、伽玛函数、贝塔函数、椭圆函数等；包括各种数学运算功能，如数值微分、数值积分、插值、求极值、方程求根、FFT、常微分方程的数值解，残数计算等等。也就是说，MATLAB 给使用者提供了足够多的数学工具，以便他们来解决各种各样的理论问题和工程问题。

2. 具有很好的绘图功能，可方便地画出两维和三维图形，如直角坐标曲线图、极坐标曲线、直方图、对数坐标曲线、网格图、三维曲面图、等高线图、区域图等，同时屏幕上出现的各种图形均可输送到打印机直接打印，或输送 Windows 的剪贴板内供各种应用软件调用。

3. 使用方便，便于掌握，便于修改，使用 MATLAB 语言编写的程序可直接运行而无须

进行编译。具有很友好的图形界面，且使用者可根据自己的需求方便地编写出自己所希望的图形界面。扩充性能很好，使用者可使用此语言编写出自己所需的各种函数，同时采用一定的措施之后可以直接调用 C 语言的函数及 FORTRAN 语言的子程序。

4. 具有若干功能强大的应用工具箱，如信号处理、通信、DSP、小波变换、图象处理、统计、系统仿真、插值、符号运算、系统识别、系统优化、神经网络、数据库、功率系统、地图、偏微分方程、系统控制、模糊逻辑、μ 分析与综合等。

5. 可以直接处理计算机内的声音文件（.wav 文件）；安装图象处理工具箱后，可以直接处理各种格式的图形文件，如：bmp、gif、pcx、tif、xwd 和 jpeg 等等。

由于 MATLAB 语言本身所具有的种种特点，它特别适合对电子系统进行仿真。特别是信号处理、DSP、通信等工具箱内，设有专门为电子系统设计的各种专门的函数，可以进行各种模拟和数字滤波器的设计和分析，如巴特伍兹滤波器、切比雪夫滤波器、椭圆滤波器，升余弦滚降滤波器等；可以分析一个系统传递函数的幅频特性和相频特性，可以对数字序列进行数字滤波；可以进行各种变换，如 Z 变换、傅氏变换、拉氏变换、希尔伯特变换等；可以实现频域或时域内的各种窗函数，如汉宁窗、汉明窗、三角窗、矩形窗等。而在使用 DSP 工具箱时，只需在图形界面上用鼠标将各个单元（如信号发生器、示波器、滤波器、乘法器、窗、频谱分析仪）连接在一起，就可看到系统的实时响应，而无须自己动手进行编程。

MATLAB 的命令又称为函数，其名称的后缀为 .m。MATLAB 语言采用解释运行的方式，在这一点上它的运行方式与 BASIC 语言类似，在程序运行的过程中可以随时显示中间结果，这样便于查找程序中的错误。另外由于用 MATLAB 语言编写的程序无需进行编译，因此它本身就象一个超级的函数计算器。由于很多 MATLAB 语言的命令与人们通常的书写习惯类似，因此有人又称之为演算纸式的科学工程计算语言。

MATLAB 语言的另一个突出的优点就是便于学习、容易掌握。一般来说，一个初学者可在几十分钟之内学会并掌握它的基本操作命令，进而就可以解决一些比较繁琐的数学运算问题，如矩阵求逆。由于 MATLAB 给使用者提供了极为丰富的、现成的数学工具，因此使用者无需掌握很复杂的编程技巧，诸如排队、指针、堆栈等等。另外，MATLAB 本身提供了十分清楚明确、十分详细的联机帮助文件，故便于使用者进行自学，而无需依赖教师指导。

1.3 MATLAB 的安装与使用

MATLAB 的安装是比较简单的，与一般计算机软件的安装过程没有什么区别。由于 MATLAB 的各个版本之间存在着差别，同时存在着二十多个用 MATLAB 语言编写的工具箱，因此安装 MATLAB 所需的硬盘空间彼此相差是很大的。

1.3.1 MATLAB 3.5

MathWorks 公司于 1989 年 1 月推出了 MATLAB 3.5 版，它是工作在 DOS 环境下的，对系统的要求是：386 以上的 IBM 兼容计算机、2MB 内存、1.6MB 的硬盘空间（包括了 Signal 工具箱）。

MATLAB 3.5 的安装盘仅为一张 3.5 英寸软盘。具体的安装过程是：首先建立一个

MATLAB 子目录，然后将安装盘放入软盘驱动器，接着键入 install 命令即可。

在安装完毕后，进入 MATLAB 子目录键入 matlab 命令，于是在屏幕的左端就会出现 MATLAB 特有的提示符》，这就表示已经进入了 MATLAB 的工作环境，然后就可以着手进行 MATLAB 的操作了。MATLAB 的工作环境与 DOS 环境十分近似，直接键入 MATLAB 语言的命令，就可以得到相应的结果。键入 quit 命令就退出了 MATLAB 的工作环境。

使用 3.5 版的 MATLAB 已经可以解决许多实际问题了，如数字滤波器的设计、对信号进行数字滤波、使用 FFT 进行频谱分析，函数绘图，线性代数的运算，代数方程的数值解等等，同时它还设有比较简单的用户图形界面，便于进行交互式的操作。笔者在九十年代中期，曾使用 3.5 版的 MATLAB 进行过有线电视系统非线性失真的分析、电视接收天线参数曲线的绘制、电波快速衰落的分析等工作。

1.3.2 MATLAB 4.2

MathWorks 公司于 1994 年推出了 MATLAB 4.2 版，它是在该公司 1992 年推出的 4.0 版的基础之上经过完善面形成的，工作在 Windows 环境下，与以前的 DOS 版本相比较，该版本的功能增加了许多，设置了友好的图形用户界面，使用起来十分方便，同时与之配套的工具箱数量达二十多个，以适应不同领域的要求。Windows 版 MATLAB 的推出，大大地拓宽了 MATLAB 的使用范围。

4.2 版的 MATLAB 本身（不包括工具箱）对系统的基本要求是：486DX 以上的 IBM 兼容计算机、4MB 内存、13MB 的硬盘空间。

MATLAB 4.2 版的安装盘有软盘和光盘两种方式。

安装盘的软盘共有 5 张，将 1 号安装盘放入软盘驱动器，用鼠标点击 setup，然后根据提示依次放入其它的安装盘即可完成整个安装过程。

安装光盘内一般都包括了很多 MATLAB 的工具箱，其数量在数个至二十多个之间，使用光盘安装 MATLAB 显然要方便得多。对于进行电子系统仿真工作来说，并不需要安装所有的工具箱，一般只需安装 Signal，Simulink，DSP，Images，Symbolic，Comm 等几个工具箱即可，这样所需的硬盘空间大概为 50MB 左右。若将二十多个工具箱全部需要约 80MB 的硬盘空间。

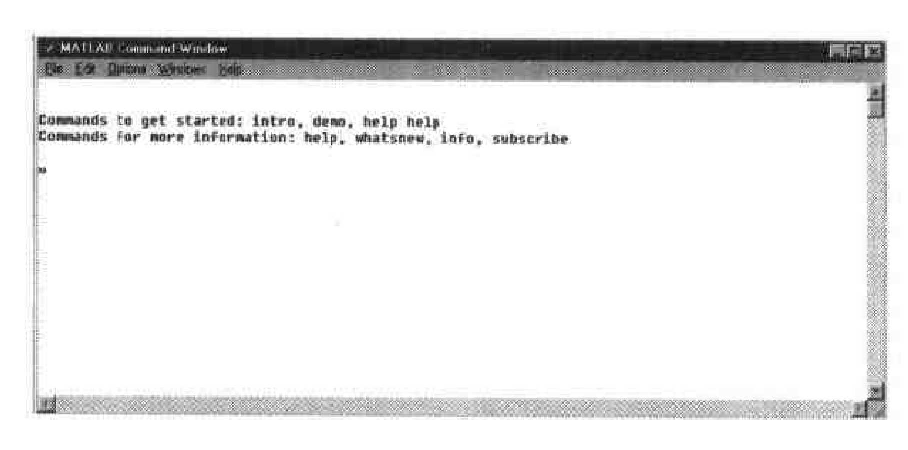


图 1.3 MATLAB 4.2 的命令窗口

安装结束之后，在 Windows 开始菜单的任务栏内点击 Matlab 图标，便可立即进入 MATLAB，屏幕上会出现图 1.3 所示的命令窗口，注意：在使用中文 Windows 操作系统时，在该窗口内不会出现 MATLAB 的提示符》。MATLAB 4.2 确实是在 Windows 环境下工作的，但是在 MATLAB 自己的工作环境内，其运行方式仍跟 DOS 系统类似，也是通过键盘键入各种命令来工作的。初学者可键入 demo 命令来启动演示程序，通过这个演示程序对 MATLAB 的功能可以有个比较全面的了解。通过 MATLAB 窗口菜单栏中的 Help 菜单可以启动联机帮助，通过它可以查询全部的 MATLAB 命令的作用和格式，实际上这是学习 MATLAB 使用的一种方便且快捷的方法。

通过菜单栏中的 File 菜单，可以完成打开 M 文件、启动程序编译器、打印、运行、退出 MATLAB 等任务，如图 1.4 所示；而通过 Edit 和 Options 菜单，则可以完成特定的编辑任务，可以选择数据的格式、字体。

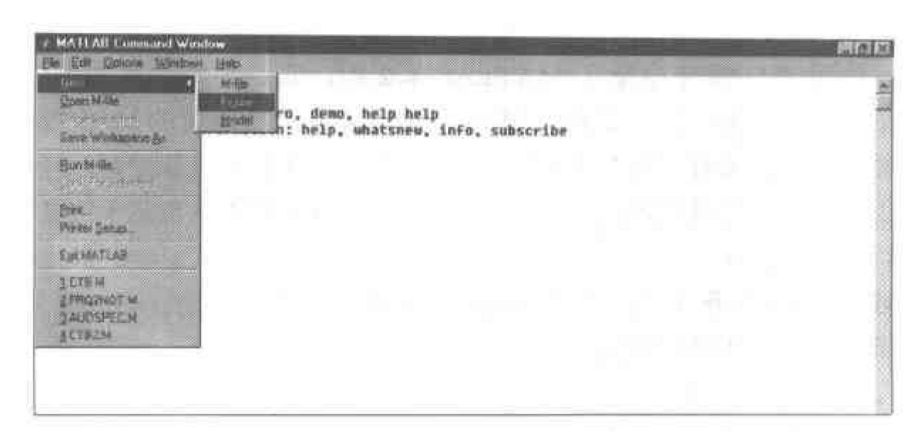


图 1.4　MATLAB 4.2 的 File 菜单

1.3.3　MATLAB 5.2

MathWorks 公司于 1998 年 1 月推出了 MATLAB 5.2 版，它在 4.X 版本的基础之上作了很大的改动，如增加了很多函数，提高了某些函数的运算速度，改变了图形窗口的外观，新增了一个功能完备的程序编辑器，设置了 PDF 格式和 HTML 格式的帮助文件等等。当前，5.2 版已经成为 MATLAB 的主流版本。

MATLAB 5.2 版的安装盘为光盘。

安装 5.2 版的 MATLAB（不包括工具箱）需要 53MB 的硬盘空间。为了进行电子系统的仿真工作来说，应该安装 Signal，Simulink，DSP，Images，Symbolic，Comm 等几个工具箱（不包括 PDF 格式和 HTML 格式的帮助文件），这样所需的硬盘空间大约为 110MB 左右。若将二十多个工具箱全部安装，不包括上述两种格式的帮助文件，需要约 140MB 的硬盘空间。

进行电子系统的仿真，在硬盘比较大的情况下，最好安装 5.2 版的 MATLAB。

1.3.4　MATLAB 5.3

MathWorks 公司于 1999 年 1 月推出了 MATLAB 5.3（R11）版，这是目前见到的最新版本。

安装5.3版的MATLAB（不包括工具箱）需要76MB的硬盘空间。为了进行电子系统的仿真工作来说，应该安装Signal，Simulink，DSP，Images，Symbolic，Comm等几个工具箱（不包括PDF格式和HTML格式的帮助文件），这样所需的硬盘空间大约为165MB左右。若将二十多个工具箱全部安装，不包括上述两种格式的帮助文件，需要约234MB的硬盘空间。

1.3.5 MATLAB的使用

进入MATLAB的工作环境之后，通过键盘输入MATLAB命令，便可以执行相应的操作了这与DOS的操作十分接近。例如，键入ver命令，屏幕上显示出MATLAB及已安装的工具箱的版本，显示的内容如下：

```
》ver
MATLAB Version 4.2c.1
MATLAB Toolbox Version 4.2a 25 - Jul - 94
Communications Toolbox MATLAB function library.     Version 1.0a 25 - Apr - 96
Communications Toolbox SIMULINK S - functions.      Version 1.0a 25 - Apr - 96
Communications Toolbox SIMULINK block library.      Version 1.0a 25 - Apr - 96
Symbolic Math Toolbox.   Version 1.1a 24 - jul - 95
DSP Blockset.   Version 1.0a 3 - Nov - 1995.
Image Processing Toolbox.   Version 1.0b 31 - Jan - 94
Signal Processing Toolbox.   Version 3.0b 10 - Jan - 94
SIMULINK model analysis and construction functions.   Version 1.3c 15 - August - 94
SIMULINK block library.   Version 1.3c   15 - August - 94
SIMULINK demonstrations and samples.   Version 1.3c   15 - August - 94
```

键入version命令，则显示MATLAB的版本。

```
》version
ans =
4.2c.1
```

在MATLAB的工作环境下可以执行DOS的命令，只需在该命令前先键入！。例如给一个文件改名，可以进行如下的操作：

```
》! ren tmp.m tmp1.m
```

有些MATLAB的命令在形式上与DOS命令相同，如：cd命令用于改变当前的工作目录，而dir命令则用来显示当前目录所有文件和子目录。这两个命令使用的频率是很高的，因为一般的工作程序是：先建立一个使用者自己的子目录，然后进入该目录内进行MATLAB的操作。

使用help命令可以查看某个MATLAB命令（函数）的具体使用方法，还可以显示某个工具箱内所有函数的名称和功能，这种方法是学习MATLAB使用的捷径。

例如键入help ver，屏幕上的显示内容为：

```
VER MATLAB, SIMULINK, and TOOLBOX version information.
VER displays the current MATLAB and toolbox version numbers.
```

VER (TBX) displays the current version information for the
toolhox specified hy the string TBX.

For example,
ver control
returns the version info for the Control System Toolbox.

See also VERSION, HOSTID, LICENSE, INFO, WHATSNEW.

使用 lookfor 命令可以进行关键词检索，这样便于查找使用者所需要的函数名称。例如某人要进行积分，但并不知道应该使用什么 MATLAB 命令，于是他就可以键入 lookfor integral，屏幕上则显示出 MATLAB 和各工具箱内与积分有关的函数名称：

ELLIPKE Complete elliptic integral.

EXPINT Exponential integral function.

DBLQUAD Numerically evaluate double integral.

INNERLP Used with DBLQUAD to evaluate inner loop of integral.

QUAD Numerically evaluate integral, low order method.

QUAD8 Numerically evaluate integral, higher order method.

COSINT Cosine integral function.

SININT Sine integral function.

ASSEMA Assembles area integral contributions in a PDE problem.

COSINT Cosine integral function.

FOURIER Fourier integral transform.

IFOURIER Inverse Fourier integral transform.

SININT Sine integral function.

使用 delete 命令可以删除文件，clc 命令的功能是清屏（清除命令窗口上显示的全部内容），而 path 命令则是对 MATLAB 的搜索路径进行控制。

参考文献

[1] 韩慧君．系统仿真．北京：国防工业出版社，1988

[2] (美) Duane Hanselman, Bruce Littlefield. 精通 MATLAB. 西安：西安交通大学出版社，1988

[3] 张培强．MATLAB 语言．合肥：中国科技大学出版社，1995

[4] 薛定宇．控制系统计算机辅助设计．北京：清华大学出版社，1996

第二章 MATLAB的基本操作

本章以4.2版本为基础，简要地介绍MATLAB的基本操作，由于较高的版本是向下兼容的，因此这些内容对于5.2版或5.3版也是有效的。对MATLAB有所了解的读者，可跳过本章而直接阅读第三章的内容。

MATLAB的操作采用命令行的方式，在一行之内可以输入多条MATLAB命令，而每条命令之间用逗号或分号隔开。采用逗号时，在屏幕上显示出每条命令的执行结果，而采用分号时，则不显示中间结果。在调试程序的过程中，显示中间结果便于使用者对程序的结构和正确性进行检查。

%（百分号）为注释行的标记，运行时编译器忽略其后的内容。

…为续行标记，它表示下面的一行与前一行是连接在一起的。

MATLAB中最主要的组成部分是函数，其一般形式为：

[输出变量列表] =函数名（输入变量列表）;

2.1 变量

2.1.1 变量的类型

MATLAB的基本操作是进行矩阵运算，并且支持复数，同时也支持字符，也就是说，MATLAB的基本变量为复数矩阵和字符矩阵。一维的矩阵就是向量（亦可称为数组），向量又有行向量和列向量之分；通常所说的字符串在MATLAB中就是字符行向量。1×1的矩阵为标量，也就是一个数（实数或复数），而1×1的字符矩阵则是一个字符。

变量的名称必须以字母开头，其后可以是任意的字符，变量名的长度不不能超过19个字符，并且区分大写和小写。

MATLAB中有几个特殊的变量，见表2.1。

表2.1 特殊变量表

特殊变量	取　值
ans	计算结果的缺省变量名
pi	圆周率 π
eps	相对浮点精度，eps=2.2204e-16
inf	无穷大
NaN	不确定量
i或j	虚数单位 $\sqrt{-1}$
nargin	函数的输入变量数目
nargout	函数的输出变量数目

对变量不需要进行说明，每个变量的类型根据所赋的值来定。变量的名称应该尽可能采用与实际问题中相同或相近的名称，如用 **omega** 表示 ω，用 **alfa** 表示 α 等等，变量名称的意义应该明确，这样便于对程序的阅读和理解，同时注意避免使用 MATLAB 中规定的特殊变量名。

2.1.2 数据显示格式

在缺省的情况下，整数的显示格式就为整数；而实数则根据数据的有效位数，取小数点后的 4 位或采用指数的显示方式。

使用 format 命令可以人为改变数据的显示格式，MATLAB 的数据格式见表 2.2。

表 2.2 数据显示格式

命　令	说　明	圆周率
format short	5 位有效数字	3.1416
format long	15 位有效数字	3.14159265358979
format bank	小数点后 2 位有效数字	3.14
format short e	5 位有效数字加指数	3.1416e+000
format long e	16 位有效数字加指数	3.141592653589793e+000
format rat	有理数近似（近似分数）	355/113
format hex	16 进制	400921fb54442d18

使用 4.2 版时，用户可在 Options 菜单中的 Numeric Format 选项中改变数据的显示 格式；而使用 5.2 版或 5.3 版时，则可在 File 菜单中找到 Preferences 项，然后在 General 子菜单中的 Numeric Format 选项中改变数据的显示格式。

2.1.3 向量

向量又称为一维数组，在编制仿真软件的过程中要经常使用它。向量有行向量和列向量之分，它们的赋值方法略有区别。

· 行向量的赋值

》a= [1 2 3 7 6 8]; 　　输入一个 6 个元素的行向量。

》b=1: 2: 99; c=2: 2: 98; 　　b 为 100 以内所有的奇数组成的行向量，而 c 为 100 以内所有的偶数组成的行向量。

应该尽量采用循环的方式进行向量的赋值，其中第一项为向量的第一个数，冒号后的第二项表示增量，第三项则为向量的最后一个数，不给定增量时，缺省的增量数值为 1。

》d=1: 100; 　　d 为 100 以内所有自然数组成的行向量。

向量的赋值也可以采用添加的方式，如：

》e= [1: 2: 9　2: 2: 10]; 与》e= [1 3 5 7 9 2 4 6 8 10]; 是相同的。

· 列向量的赋值

列向量的赋值可以采用行向量转置的方式。

》e= (1: 2: 7)′

e=

1

3

5

7

· 向量的运算

在MATLAB中分别定义了数与向量间的运算，以及向量之间的运算。

向量与数之间的运算是针对每个向量元素进行的，如：

》a=1: 2: 9; b=2: 2: 10; c=2 * a; d=b-1.5; e=a/3;

》x=0: .01: 10; x=x * pi; 表示x的取值范围在0和10π之间，增量为0.01π。

在两个向量之间进行运算时，它们的长度应该相等，也就是说它们的元素数目应该相同；另外进行向量元素的乘、除、乘方运算时，运算符前要加上一个“.”。如：

》a=1: 2: 9; b=2: 2: 10; c=a+b; d=a. * b; e=1./a; f=a./b; g=a.^2;

若》b=1: 2: 9; c=2: 2: 8; 由于b和c两个向量的长度不同，故两者不能直接进行运算，但是可以采用以下的运算方式。

》d=b (1: 4) +c; 表示b向量的前4个元素与c向量相加。

length函数可以确定一个向量的大小，该函数的使用方法是：

》m=length (a); m为向量a的长度

》f= []; m=length (f); f为一个空向量，m的值为0

· 矢量运算

矢量是向量的特例，矢量运算主要是数量积（点积）和矢量积（叉积）两种。dot函数的功能是点乘，cross函数的功能是叉乘。

》A= [1 2 3]; B= [2 0 1];

》D=dot (A, B); C=cross (A, B); C1=cross (B, A)

运算结果为 D: 5, C: [2 5 -4], C1: [-2 -5 4]

2.1.4 复数

复数作用在电子系统仿真中是十分重要的，在MATLAB中它的表示与一般数学书中的表示方法是一样的。

· 复数的赋值

复数的赋值采用实部系数和虚部系数的方法，而虚数则用虚数单位来表示，如

》a=1+2j; b=3+4i; c= [a b 5-2j 3+1j];

· 复数的运算

复数的四则运算和乘方运算的方式与实数是完全一样的，另外复数还有其特有的运算方

式。

模	abs (1+2j)	2.2361
幅角（单位为弧度）	angle (1+2j)	0.9273
实部	real (1+2j)	1
虚部	imag (1+2j)	2
共轭	conj (1+2j)	1.0000-2.0000i

2.1.5 矩阵

矩阵是 MATLAB 的基本运算单元，矩阵也被称为二维数组。最初开发 MATLAB 的目的就是为了简化矩阵和线性代数繁琐的运算过程。在 5.2 版以上的 MATLAB 中，还支持三维矩阵（三维数组）。

·矩阵的赋值

矩阵的赋值一般采用逐行赋值的方式，如：

》A= [1 2 3; 4 5 6; 9 7 8];　　定义了 A 为 3×3 矩阵 $\begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\\9 & 7 & 8\end{pmatrix}$

在矩阵中冒号是很有用的，如 A (:, 2) 表示 A 矩阵的第二列（列向量），而 A (3, :) 则表示该矩阵的第三行（行向量）。

若想将矩阵 A 转换为向量，则可以键入 a=A (:)；转换是按列进行的，因此 a 也是列向量，它共有 9 个元素。

·矩阵的运算

矩阵的运算完全是按照线性代数的运算规则而进行的。

```
》A= [1 2 3; 4 5 6; 9 7 8]; B= [1 2 0; 0 0 1; 3 1 2];        % 矩阵
》c= [1 1 2];    d= [3 2 1]';                                 % 向量
```

矩阵的加法　　D1=A+B;　　E1=A-B;

矩阵的数乘　　D2=2*A;　　E2=B/3;

矩阵的乘法　　D=A*B;　E=A*d;　F=c*A;　G=d*c;　H=c*d;

运算结果为 D: $\begin{pmatrix}10 & 5 & 8\\22 & 14 & 16\\33 & 26 & 23\end{pmatrix}$, E: $\begin{pmatrix}10\\28\\49\end{pmatrix}$, F: (23　21　25), G: $\begin{pmatrix}3 & 3 & 6\\2 & 2 & 4\\1 & 1 & 2\end{pmatrix}$, H: 7

矩阵转置（非共轭转置）　　C=A.';

矩阵转置（共轭转置）　　D=A';

逆矩阵　　E=inv (A);

下面给出两个矩阵运算的实例。

例 1　三元天线阵的电流分布

三个半波振子组成的三元天线阵，如图 2.1 所示，设每个天线的激励电压均为 1V，求每个半波振子的输入电流强度。

使用 MATLAB 解决这种问题是十分简便的，具体步骤是：先确定天线阵的阻抗矩阵，然后求其导纳矩阵，而导纳矩阵与激励电压向量的乘积就是所求的电流向量。

阻抗元素的数值可以从天线的互阻抗曲线查到。以下为程序清单和计算结果：

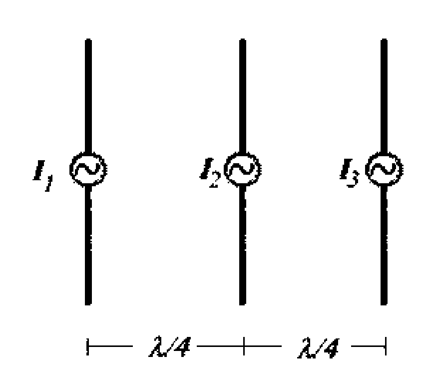


图 2.1 三元天线阵

```
》z11=73+43j; z12=40-30j; z13=-18-25j;
》Z= [z11 z12 z13; z12 z11 z12; z13 z12 z11];
》I=inv (Z) * [1 1 1]';
 I=
    0.0090 - 0.0019i
    0.0082 + 0.0047i
    0.0090 - 0.0019i
```

例 2 NTSC 的亮度和色度信号

若颜色为饱和度为 50%的品红，即 R=1，G=0.5，B=1，亮度信号 Y 可以由亮度方程 Y = 0.3R + 0.59G + 0.11B 给出，而色度信号为

$$\begin{bmatrix} I \\ Q \end{bmatrix} = \begin{bmatrix} \cos 33^\circ & -\sin 33^\circ \\ \sin 33^\circ & \cos 33^\circ \end{bmatrix} \begin{bmatrix} 0.877(R - Y) \\ 0.493(B - Y) \end{bmatrix}$$

程序清单和计算结果为：

```
》R=1; G=.5; B=1; Y=.3*R+.59*G+.11*B;
》T=[cos(33*pi/180)-sin(33*pi/180); sin(33*pi/180)cos(33*pi/180)];
》C=T* [.877* (R-Y) .493* (B-Y)]'; I=C (1); Q=C (2);
Y = 0.7050,    I = 0.1378,    Q = 0.2629
```

由于 MATLAB 中三角函数均采用弧度计算，因此必须将角度先化成弧度，然后计算三角函数的数值。

另外在 MATLAB 中还定义了矩阵元素之间的运算，这与向量元素间的运算是类似的，在矩阵元素作乘法、除法和乘方运算时，在运算符前要加上一个“.”。

size 函数可以确定一个矩阵的大小，该函数的使用方法是：

```
》[m, n] =size (A);    m为A矩阵的行数，n为A矩阵的列数
```

· 空矩阵

空矩阵就是 0×0 矩阵，A= []；就定义了一个空矩阵。

· 常数矩阵

常数矩阵有 1 矩阵、0 矩阵和单位矩阵三种。

ones 函数将所有的矩阵元素赋值为 1。

```
》B=ones (2, 3); C=ones (4, 3) *2;
        B为2行3列的1矩阵，C为4行3列的常数矩阵，其元素均为2
》B=ones (size (A));        B为1矩阵，其大小与A矩阵相同
》B=ones (size (c));        若c为向量，则B为1向量，其大小与c相同
```

zeros 函数将所有的矩阵元素赋值为 0。

》B = zeros (2, 3); B 为 2 行 3 列的 0 矩阵

》B = zeros (size (A)); B 为 0 矩阵，其大小与 A 矩阵相同

》B = zeros (size (c)); 若 c 为向量，则 B 为 0 向量，其大小与 c 相同

eye 函数的功能是生成单位矩阵，它取的是英文字母 I 的谐音。

》B = eye (3); B 为 3 行 3 列的单位矩阵

2.1.6 字符向量和字符矩阵

字符向量就是通常所说的字符串，而字符矩阵则是由若干个长度相等的字符向量组成的。

· 字符向量

字符向量的赋值与普通向量类似，但要加上单引号。

》s = 'Beijing Broadcasting Institute';

s 为 30 个元素的字符向量，s (9) 为字母 B，而 s (30) 为字母 e。

》s = 'Beijing'; s = [s ' Broadcasting Institute']; 与前面的结果相同。

当在字符向量内出现单引号时，应该改为两个单引号（注意不是双引号），如：

》s = 'It''s yours.' 屏幕上显示内容为 It's yours.

·字符矩阵

字符矩阵的赋值一般使用函数 str2mat。在 MATLAB 中很多转换函数的名称中均有 2 字，它是英文单词 to 的谐音，意为转换。str2mat 表示将字符串转换成为字符矩阵。

》S1 = str2mat ('Beijing', 'Tianjin', 'Shanghai');

》S2 = ['Beijing ';'Tianjin ';'Shanghai'];

S1 与 S2 是相同的，均为 3 行 8 列的字符矩阵。函数 str2mat 自动在较短的字符行向量后面补上空格，以使每行的长度一样。

2.1.7 几个常用的操作命令

clc 命令清除当前命令窗口的所有显示内容。

clear 命令用于清除当前工作空间内的变量。

》clear; 清除当前工作空间内所有的变量

》clear A a b; 清除变量 A、a 和 b，保留其它的变量

who 命令用于查询当前工作空间内所有变量的名称；而 whos 命令则显示出当前工作空间内所有变量的名称、大小、字节、是否为复数等信息。

which 命令显示出某个 MATLAB 函数的路径。

》which cross

h: \ matlab42 \ toolbox \ matlab \ datafun \ cross.m

2.2 关系和逻辑运算

关系运算和逻辑运算在一个仿真软件中的作用是十分重要的。在MATLAB中，对于所有关系和逻辑表达式的输入参数来说，任何非零数值代表“真”，而0则代表“假”；而对于关系和逻辑表达式的输出来说，1表示“真”，而0则表示“假”。

· 关系操作符

表2.3给出了MATLAB中的关系操作符。

表2.3 关系操作符

关系操作符	含 义
<	小于
<=	小于等于
>	大于
>=	大于等于
==	等于
~=	不等于

由于关系运算的结果均为数量，不是1，就是0，因此灵活地使用关系表达式，可以简化某些运算的形式，下面举例说明。

例3 避免0作除数

```
》x= (-3: 3) /3;    y=sin (x) ./x
Warning: Divide by zero
y=
    0.8415   0.9276   0.9816     NaN   0.9816   0.9276   0.8415
```

由于向量x中有一个元素为0，在计算函数sin (x) ./x的过程中，出现了除数为0的情况，因此向量y中的第四个元素为不确定量（NaN），为了避免函数中出现这种没定义的情况，可利用计算机中的相对浮点精度eps来取代0，具体做法如下：

```
》x= (-3: 3) /3;    x=x+ (x==0) *eps;    y=sin (x) ./x
y =
    0.8415   0.9276   0.9816   1.0000   0.9816   0.9276   0.8415
```

其中关键的是引入了关系表达式x==0，其运算结果为向量[0 0 0 1 0 0 0]，然后将这个向量先与eps相乘，然后再与x向量相加，这样就利用eps取代了0。

例4 单位采样序列δ ($n-n_0$)

单位采样序列是离散信号中的一个数学模型，其定义为

$$\delta(n-n_0)=\begin{cases}1, & n=n_0\\0, & n\neq n_0\end{cases}$$

```
》n=-5: 5;    del= (n-2==0);    [n; del]
ans =
```

```
   -5    -4    -3    -2    -1    0    1    2    3    4    5
    0     0     0     0     0    0    0    1    0    0    0
```

实现单位采样序列的命令只有 2 条，第三条命令只是为了显示运算结果，其结果完全符合要求。

例 5　单位阶跃序列 $u(n-n_0)$

单位阶跃序列也是离散信号中的一个数学模型，其定义为

$$u(n-n_0)=\begin{cases}1, & n\geqslant n_0\\ 0, & n<n_0\end{cases}$$

```
》n=-5: 5; u= (n+1>=0);     [n; u]
  ans =
  -5    -4    -3    -2    -1    0    1    2    3    4    5
   0     0     0     0     1    1    1    1    1    1    1
```

实现单位阶跃序列的命令也只有 2 条，其结果完全符合要求。

· 逻辑运算符

表 2.4 给出了 MATLAB 中的逻辑运算符。

表 2.4　逻辑运算符

逻辑运算符	含　　义
&	与
|	或
~	非

```
》n=1: 11; k= (n>2) & (n<8)
k =
    0    0    1    1    1    1    1    0    0    0    0
》n=1: 11; k=~ (n>5)
k=
    1    1    1    1    1    0    0    0    0    0    0
```

· 逻辑运算函数

xor (x, y) 函数：异或运算，x 或 y 非零（真）返回 1，x 和 y 都是零返回 0，x 和 y 都非零（真）返回 0。

```
》xor ( [1 1 0 0], [0 1 0 1])
ans =
    1    0    0    1
```

all 函数：所有向量元素（对于矩阵来说则指列向量）非零时，输出为 1；否则，输出为 0。

```
》all (1: 9);                    输出为 1
```

》all ([1: 9 0]); 输出为 0

》all ([1: 3; 1: 3; 2: 4]); 输出为 [1 1 1]

》all ([1: 3; 1: 3; 0: 2]); 输出为 [0 1 1]

any 函数：向量中有非零元素（对于矩阵来说则指列向量）时，输出为 1；否则，输出为 0。

》any ([zeros (1, 9) 1]); 输出为 1

》any (zeros (1, 9)); 输出为 0

》any (zeros (3, 3)); 输出为 [0 0 0]

》A = zeros (3, 3); A (3, 2) = 1; any (A); 输出为 [0 1 0]

· 测试函数

在 MATLAB 中提供了一批测试函数，用来判断某些条件是否成立，其输出为逻辑值，见表 2.5。

表 2.5 一些测试函数

测试函数	功　能	说　明
exist ('A')	变量或函数是否存在	A 不存在为 0，A 是变量为 1，A 是函数为 2
finite	元素是否为有限值	finite ([pi NaN Inf]) – [1 0 0]
isempty	是否为空矩阵	isempty ([ ]) – 1，isempty (1: 9) – 0
isglobal ('a')	是否为全程变量	a 为全程变量时输出为 1，否则为 0
isinf	是否为无穷大	isinf ([Inf 3]) – [1 0]
isnan	是否为不确定值	isnan ([NaN 9]) – [1 0]
isreal	是否为实数	isreal ([1 – 1j pi]) – 0，isreal (1: 9) – 1
isspace	是否为空格字符	isspace ('It is.') – [0 0 1 0 0 0]
isstr ('S')	是否为字符变量	S 为字符向量或字符矩阵时为 1，否则为 0

2.3 条件执行语句

任何程序设计语言中都必须设有条件执行语句，根据条件来选择执行相应的程序段。在 MATLAB 设有两种类型的条件执行结构。

2.3.1 if – else – end 结构

if – else – end 结构是最基本的条件执行语句，用 if 开头，以 end 结尾，这两条是必不可少的。if – else – end 结构的一般形式如下：

```
if 条件表达式;
    程序段 1        % 若条件成立，则执行程序段 1
else;
    程序段 2        % 若条件不成立，则执行程序段 2
end;
```

注意：在有些情况下，else 是可以没有的，如：

```
if 条件表达式;
    程序段          % 若条件成立，则执行此程序段；若条件不成立就不执行
end;
```

2.3.2 if - elseif - end 结构

if - elseif - end 结构的作用类似一个多路开关，根据条件在多个程序段中进行选择。

```
if 条件表达式 1;
      程序段 1              % 若条件 1 成立，则执行程序段 1
elseif  条件表达式 2;
      程序段 2              % 若条件 2 成立，则执行程序段 2
elseif  条件表达式 3;
      程序段 3              % 若条件 3 成立，则执行程序段 3
else;
      程序段 4              % 若上述条件都不成立，则执行程序段 4
end;
```

2.4 循环

循环是任何程序设计语言中都必须具备的环节。在 MATLAB 设有两种类型的循环方式。由于在 MATLAB 中循环运算速度是比较慢的，因此应该尽量避免使用循环。

2.4.1 for 循环

在 for 循环中，定义了一个循环变量。采用初值、循环步长和终值来确定循环变量的取值范围，循环步长可以是小数，可以是负数，其缺省值为 1。在循环的过程中，循环变量的数值自动改变。for 循环的循环次数是可以确定的。for 循环（单循环）的结构如下：

```
for i=is: id: ie ;        % is - 初值, id - 步长, ie - 终值
    循环体
end;
```

为了提高运算速度，尽可能采用向量来完成单循环的作用。例如，下面两段程序的功能是完全一样的，而后者的运算速度快。

```
for i=1: 1000;                              %程序段 1
  x=i*.01*pi; y (i)  =sin (x) /x;
end;
x=pi* (.01: .01: 10); y=sin (x) ./x;        %程序段 2
```

多重循环的结构如下：

```
for i=i1: i2: i3;
  for j=j1: j2: j3;
    循环体
  end;
end;
```

2.4.2　while 循环

与 for 循环不同，while 循环的循环次数是不定的，它要根据条件来判断循环是否结束。while 循环的结构如下：

```
while 条件表达式；
        循环体          % 若条件成立，则执行循环；若条件不成立就跳出循环
end；
```

注意：在使用 while 循环时要特别小心，因为考虑不周，就容易形成死循环。

2.4.3　循环的中止

使用 break 语句，可以在条件成立的情况下中止循环（从循环体跳出），如：

```
for i=i1：i2：i3；
  循环体
  if 条件表达式；break；end；        % 若条件成立，则中止循环
end；
```

2.5　二维绘图

具备了便于使用且功能强大的绘图功能是 MATLAB 的主要特点之一，从某种意义上说，正是由于 MATLAB 给使用者提供了十分方便的绘图功能，因而促进了它在各个领域的推广和应用。使用二维绘图功能，可以清楚准确地显示出函数的曲线，于是分析函数的极值点、驻点以及方程求根等数学问题就变得比较容易进行了，并且其结果是十分直观的；另外在处理实验数据等方面，MATLAB 的绘图功能也给使用者提供了很大的方便。除了基本的绘图函数以外，MATLAB 还设置有很多绘图的辅助函数，熟练地使用这些辅助函数，可以绘制出十分漂亮的图形。

2.5.1　plot 函数

plot 函数是绘制二维图形的基本函数，是每个使用者都必须熟练掌握的，它的基本功能是打开一个新的图形窗口，然后在该窗口内根据数据来绘制折线；若已经存在一个图形窗口，则自动将该窗口的显示内容清除，然后再绘制新的图形。

在不同版本的 MATLAB 中，plot 函数所绘制的图形有所不同。在 4.2 版的 MATLAB 中，二维图形的背景颜色为黑色（缺省值）；而在 5.2 版和 5.3 版中，二维图形的背景颜色为白色（缺省值），而且图形还有一个灰色的边框。

plot 函数的用法是很多的，其典型用法如下：

· plot（x，y）

函数输入参数 x 和 y 为长度相同的行向量或列向量，其功能是绘制连接点（x_1，y_1），（x_2，y_2），…（x_n，y_n）之间的折线，在 4.2 版中，折线为黄颜色（缺省值），而在 5.x 版中，折线为蓝颜色（缺省值）。

例如：

》plot（[1 2]，[0 5]）　　在点（1，0）和（2，5）间画出一条线段

》x= [eps　pi*(.01：.01：10)]；plot（x，sin(x)　./x)；

在 0 至 10π 的区间内，绘制函数的曲线，步长为 0.01π

· plot（y）

若函数的输入参数 y 为一个实数向量，那么它的作用是绘制 y 向量与其下标向量构成的折线图，即 plot（y）与 plot（1：length（y），y）等价。

若函数的输入参数 y 为一个复数向量，则 plot（y）与 plot（real（y），imag（y））等价。

· plot（x，y，c）

参数 c 为字符串，使用它可以改变曲线的颜色和类型，其具体含义见表 2.6。

表 2.6　曲线的颜色与类型

字母	y	m	c	r	g	b	w	k	
颜色	黄	品红	青	红	绿	蓝	白	黑	
标点	.	o	x	+	-	*	:	-.	--
类型	点	圆圈	叉	加号	实线	星号	点线	点划线	虚线

参数 c 可以是一个字母、一个字母及一个标点、一个字母及两个标点、一个标点或两个标点，如：

》plot（x，y，'b'）；plot（x，y，'r+'）；plot（x，y，'m--'）；

》plot（x，'o'）；plot（y，'-.'）；

· plot（x1，y1，c1，x2，y2，c2）

使用给定的颜色和类型同时绘制出两条曲线，在参数 c1，c2 没有输入的情况下，曲线的颜色和类型按缺省值来定。

使用类似的方式，可以同时绘制多条曲线。

2.5.2　绘图辅助函数

在 MATLAB 中设有很多绘图的辅助函数，用于对图形进行进一步的修饰。

· title 函数

title 函数用于给图形加上标题。

title（'words'）；　　将字符串'words'添加在图形上方的中部。

· xlabel 函数

xlabel 函数用于对图形的 x 轴进行说明。

xlabel（'words'）；　　将字符串'words'添加在图形 x 轴的下面，并自动居中。

· ylabel 函数

ylabel 函数用于对图形的 y 轴进行说明。

ylabel ('words')；　将字符串'words'添加在图形 y 轴的左面。

· text 函数

text 函数用于在图形的指定位置添加文本信息。

text (x, y,'words')；　在指定位置 (x, y) 添加字符串'words'，其中 x 和 y 的单位依当前的坐标而定；在未加进一步说明的情况下（缺省值），它们代表着字符串左端的坐标。

·gtext 函数

gtext 函数的功能与 text 函数类似，只是使用鼠标来指定添加文本信息的位置。

gtext ('words')；在图形中出现一个十字线，移动鼠标寻找适当的位置，然后点击鼠标的左键，便可在该处添加字符串'words'。

· grid 函数

grid 函数用于给图形添加栅格，其后面可以带上参数 on 或 off，也可以不带参数。

》grid on;　　给图形添加栅格

》grid off;　　去除栅格

》grid;　　在添加栅格和去除栅格两种状态中，交替进行选择

· whitebg 函数

whitebg 函数用于改变图形的背景颜色，其内表示颜色的参数可以选择字符或选择表示 RGB 数值的向量。在 4.2 版的 MATLAB 中，图形背景颜色的缺省值为黑色；在 5.2 版和 5.3 版中，图形背景颜色的缺省值为白色。

》whitebg ('w')；或 whitebg ([1 1 1])；　　白色背景

》whitebg ('k')；或 whitebg ([0 0 0])；　　黑色背景

》whitebg ('r')；或 whitebg ([1 0 0])；　　红色背景

》whitebg ([.8 .8 .8])；　　灰色背景

例 6　绘制 sin (x) /x 的函数曲线

```
》x=pi* (0: .01: 6);   x=x+ (x==0) *eps; y=sin (x) ./x;
》whitebg ('w');   plot (x, y,'k'); grid;
》xlabel ('x');   ylabel ('y'); title ('sin (x) /x');
》text (3.5, -0.3,'minimum'); text (6.5, 0.17,'maximum');
```

· legend 函数

legend 函数用于给图形添加图例，可在绘制多条曲线的情况下使用。

》legend ('1st','2nd','3rd','4th')；　　按照顺序给曲线族添加图例

》legend off;　　去除添加图例的功能

例 7　绘制两条三角函数曲线

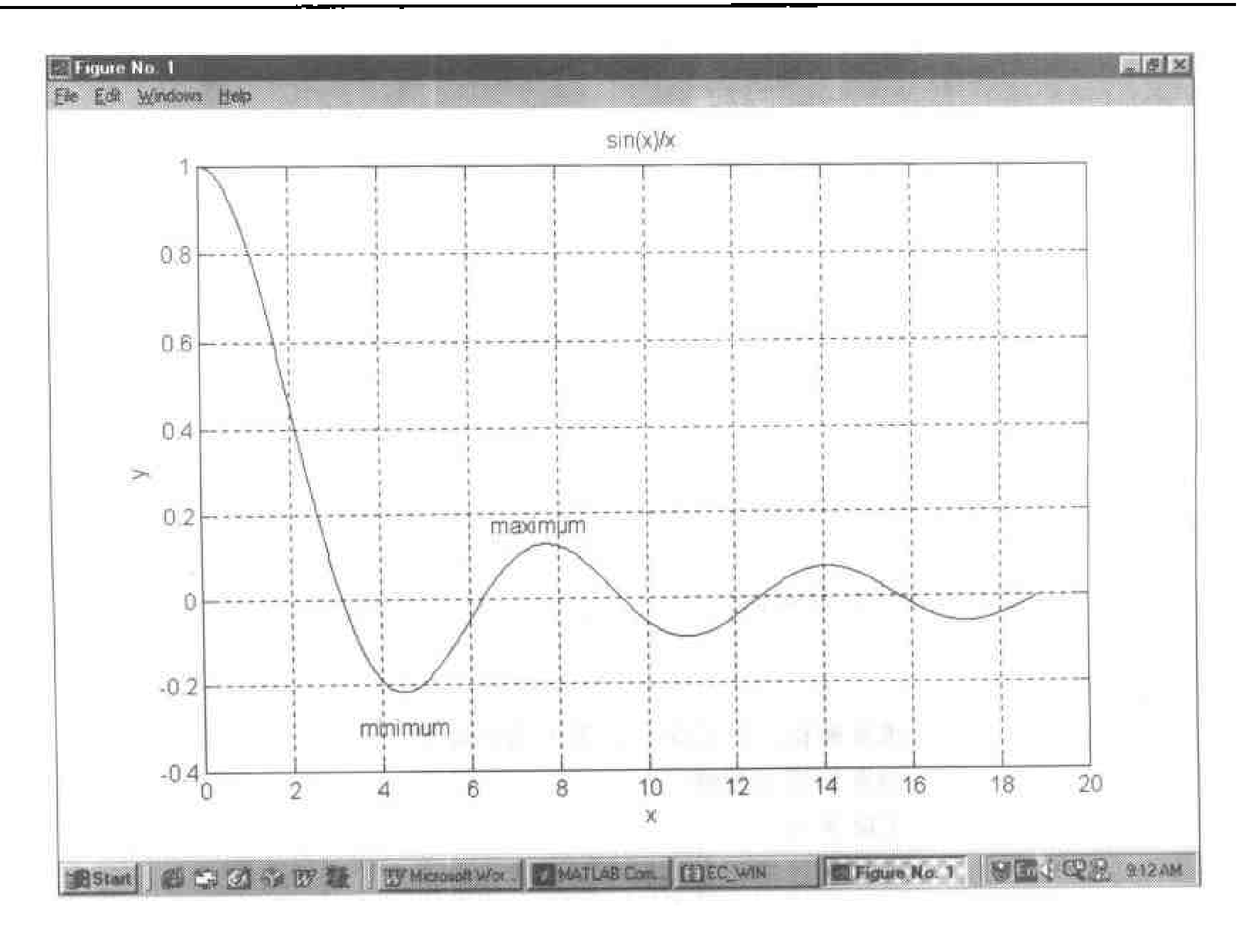


图 2.2 plot 函数的图形窗口

```
》x=pi* (0: .01: 2);    y1=sin (x);    y2=sin (2*x);
》plot (x, y1,'b', x, y2,'k:');
》legend ('sin (x)','sin (2x)');
》xlabel ('x');    ylabel ('y');
```

·hold 函数

hold 函数用于进行图形保持。使用 plot 函数绘图时，该函数会对图形窗口进行自动更新，若想在已经存在的图形上添加曲线，就需要使用 hold 函数。hold 函数可以带参数，也可以不带参数。

```
》hold on;     进行图形保持
》hold off;    去除图形保持
》hold;        在进行图形保持和去除图形保持两种状态中，交替进行选择
```

例 8 信号采样（采样时间间隔为 0.05ms）

```
》t=1e-03*(0:.005:1); x=sin(2*pi*1000*t)-.3*cos(2*pi*3000*t)+1.5;
》plot (t, x,'k:');    hold on;
》xlabel ('t (s)');    ylabel ('x (t)'); title ('Sampling');
》for i=1: 21;
  n= (i-1) *10+1;    plot ( [t (n) t (n)], [0x(n)],'b');
```

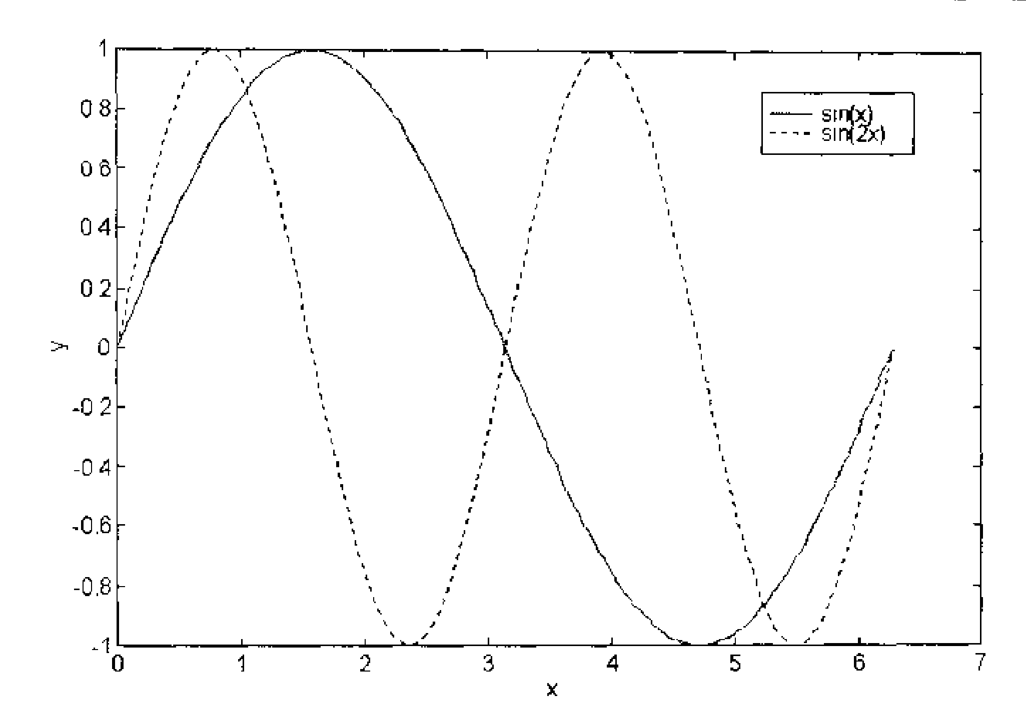


图2.3　两条正弦曲线

end; hold off;

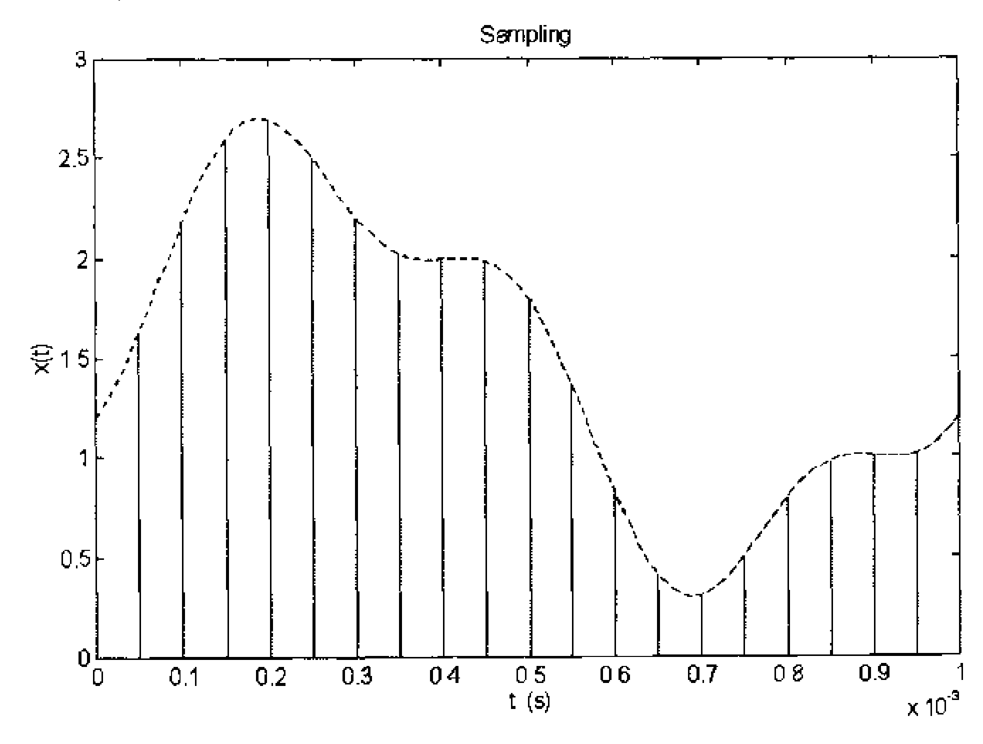


图2.4　信号采样

·axis 函数

axis 函数用于对图形的坐标进行控制，其主要用法如下：

〉axis off;　　　　　在图形上去除坐标轴的显示

〉axis on;　　　　　重新显示坐标轴

》axis equal;　　　　　　使坐标轴的长度单位相等
》axis image;　　　　　　确保图象的象素为正方形
》axis normal;　　　　　　去除 equal 或 image 的设定
》axis auto;　　　　　　恢复坐标的缺省设置
》axis ([xmin xmax ymin ymax]);　　根据给定的坐标向量，确定图象的界限
》v=axis;　　　　　　将当前坐标系的界限赋值给向量 v

例 9　在屏幕上画一个圆（若不对坐标设置进行改动，则屏幕上显示为一个椭圆）

```
》t=(0:360)*pi/180;   x=cos(t); y=sin(t);          % 圆的参数方程
》plot(x, y,'b');   hold on;
》plot([-1.5 1.5], [0 0],'r:', [0 0], [-1.5 1.5],':');      %绘制坐标轴
》axis equal; axis off; hold off;
```

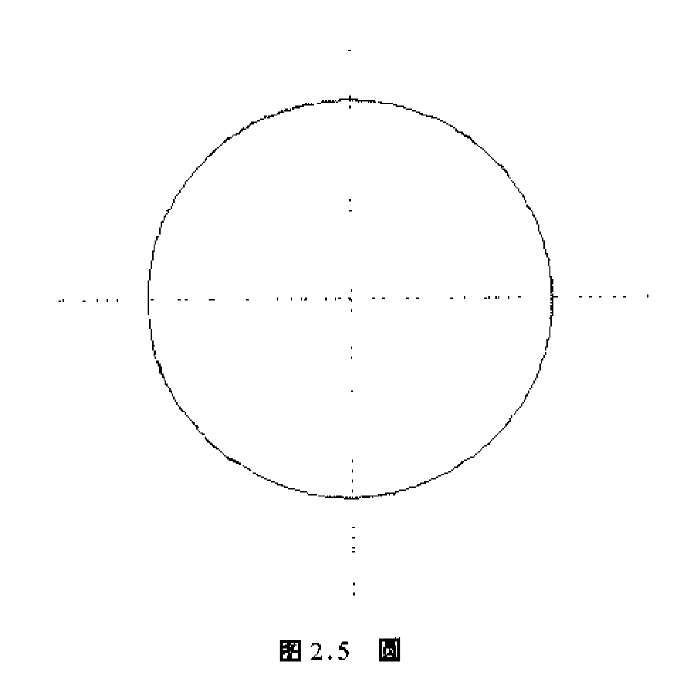


图 2.5　圆

· subplot 函数

subplot 函数的作用是在指定位置建立坐标，它与其它的绘图函数配合使用，即可在一个图象窗口内绘制多个坐标系，也就是可以绘制多幅曲线图；其基本用法是：

》subplot (mnp);　　将屏幕分割成为 m×n 个区域（m 和 n 均小于 4），而 p 则代表当前区域的序号；subplot (211) 表示屏幕上为上下两个区域，当前的图形位于上面，而下面的图形则用 subplot (212) 表示。

》subplot ('Position', [left bottom width height]);
　　根据输入的位置向量来建立坐标系

》subplot (111);　　恢复到屏幕中仅存在一个坐标系的缺省状态

在使用了 subplot 函数后，其它绘图辅助函数如 xlabel, ylabel, title, grid, axis, text 等均仅在 subplot 指定的图形区域内有效。

例 10　均匀传输线上的阻抗曲线（包括实部和虚部两部分）

```
》x=0: .01: 1; Zc=50; Zl=70;
》Z=Zc* (Zl+j*Zc*tan (x*2*pi)) ./ (Zc+j*Zl*tan (2*pi*x));
》whitebg ('w');
》subplot (211); plot (x, real (Z),'b');
》ylabel ('R (Ohm)');
》title ('Impedance of Transmission Line');
》subplot (212); plot (x, imag (Z),'b', [0 1], [0 0],'k');
》ylabel ('X (Obm)'); xlabel ('x (in wavelengtb)');
```

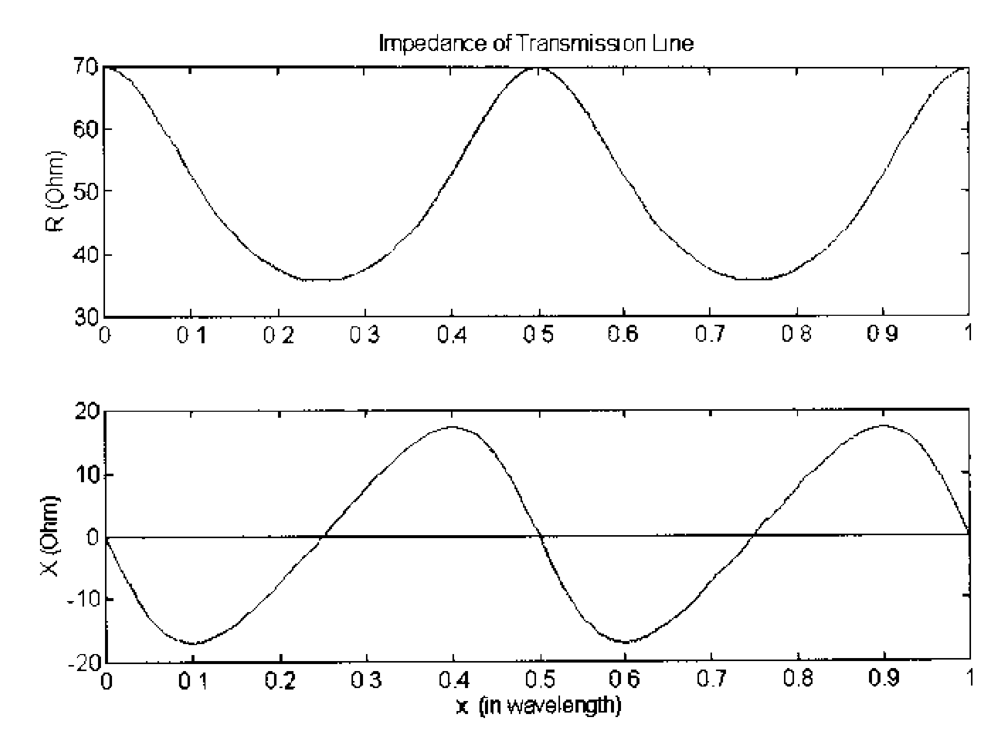


图 2.6 均匀传输线的阻抗曲线

· ginput 函数

ginput 函数多用于在曲线上读取数据，其用法如下：

》[x, y] =ginput (1);	使用鼠标的左键点击屏幕一次，返回的是点击点的坐标位置。
》[x, y] =ginput (n);	使用鼠标的左键点击屏幕 n 次，返回的是点击点的坐标位置向量。

· zoom 函数

zoom 函数用于对二维图形进行放大或缩小，这样便于对曲线进行细致的观察，其用法如下：

》zoom on;	对进行图形进行放大，使用鼠标的左键点击要进行放大的区域，放大倍数为两倍，亦可使用鼠标托曳要放大的区域，点击鼠标的右键，则图形逐步地恢复至初始状态。

》zoom xon;　　　　　仅对图形的 x 坐标进行放大，用法同上。

》zoom yon;　　　　　仅对图形的 y 坐标进行放大，用法同上。

由于 zoom、lengend、gtext、ginput 等函数均要使用鼠标，因此为了避免相互间的干扰，在使用 zoom 函数之前，应当将 lengend、gtext、ginput 函数关闭。

2.5.3　其它的二维绘图函数

plot 函数是最基本的二维绘图函数，另外 MATLAB 中还设置了一些其它的二维绘图函数，以绘制一些特定的图形。

· loglog 函数

loglog 函数采用全对数坐标（x 轴和 y 轴均为对数坐标）进行绘图，其用法与 plot 函数相同：

》loglog (x, y, c);　　　根据字符串 c 中确定的颜色和类型，采用全对数坐标绘图，若没有输入 c 参数，则按照缺省的颜色和类型绘图。

》loglog (x1, y1, c1, x2, y2, c2, x3, y3, c3);　　　采用全对数坐标同时绘制三条曲线。

· semilogx 函数

semilogx 函数采用半对数坐标（x 轴）进行绘图，其用法与 plot 函数相同：

》semilogx (x, y, c); 根据字符串 c 中确定的颜色和类型绘图，x 轴为对数坐标。

》semilogx (x1, y1, c1, x2, y2, c2, x3, y3, c3); 采用半对数坐标绘制三条曲线。

· semilogy 函数

semilogy 函数采用半对数坐标（y 轴）进行绘图，其用法与 plot 函数相同：

》semilogy (x, y, c); 根据字符串 c 中确定的颜色和类型绘图，y 轴为对数坐标。

》semilogy (x1, y1, c1, x2, y2, c2, x3, y3, c3); 采用半对数坐标绘制三条曲线。

例 11　二阶网络传递函数的幅频特性曲线和相频特性曲线

设二阶网络传递函数为：

$$H(j\omega) = 10\frac{1+(j\omega/1000)}{[1+(j\omega/100)]\cdot[1+(j\omega/10000)]}$$

```
》w=10: 10: 100000; f=w/ (2*pi);
》H=10* (1+j*w*0.001) ./ (1+j*w*.01) ./ (1+j*w*.0001);
》whitebg ('w');
》subplot (211); semilogx (f, 20*log10 (abs (H)),'b'); grid;
》ylabel ('abs (H) (dB)'); xlabel ('f (Hz)');
》title ('Amplitude and Phase Frequency Response Characteristics');
》subplot (212); semilogx (f, angle (H) *180/pi,'b'); grid;
》ylabel ('angle (H) ° '); xlabel ('f (Hz)');
```

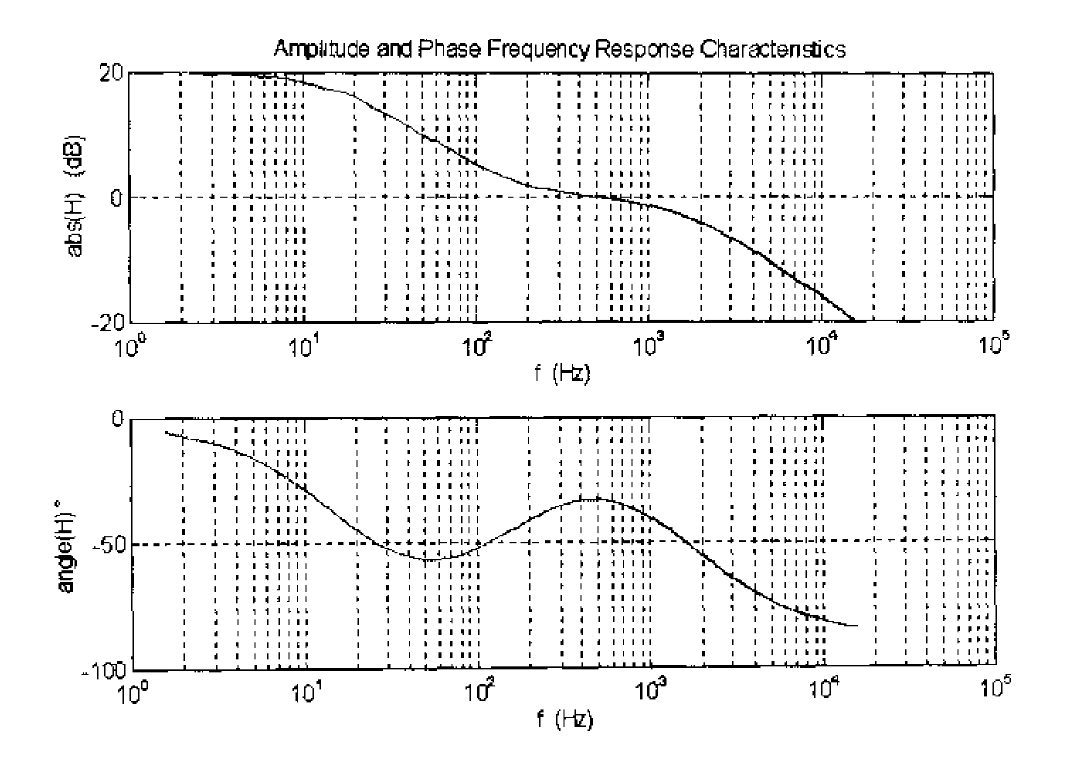


图 2.7 二阶网络传递函数的频率特性

· polar 函数

polar 函数用于绘制极坐标图，其用法是：

》polar (th, r, c)；　　根据字符串 c 中确定的颜色和类型绘制极坐标图；参数 th 为极角，单位为弧度；参数 r 则代表极径。

例 12　对称振子天线的方向图

对称振子天线的方向函数为：

$$F(\theta) = \frac{\cos(kl\cos\theta) - \cos kl}{\sin\theta \cdot (1 - \cos kl)}$$

设对称振子一臂的长度为 l=0.65λ。

```
》th= (0: .25: 180) *pi/180;    th=th+ (th==0) *eps;
》kl=2*pi*0.65;
》F= (cos (kl*cos (th)) -cos (kl)) ./sin (th) / (1-cos (kl));
》whitebg ('w');
》polar ( [th th+pi], [abs (F) ahs (F)],'b');
```

· stem 函数

stem 函数用于绘制离散序列数据图，其用法是：

》stem (y)；　　以小圆圈代表数值，并在小圆圈与 x 轴之间绘制竖线条。

》stem (x, y, c)；　　以小圈圈代表数值，并在小圆圈与 x 轴之间绘制竖线条，而竖线的类型则根据字符串 c 来确定。

例 13　单位阶跃序列 u $(n-n_0)$

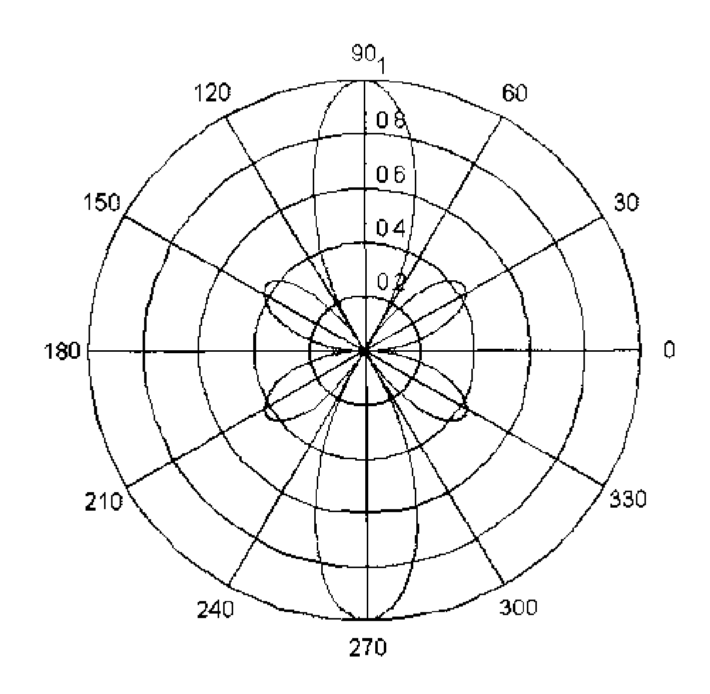


图 2.8 对称振子天线的方向图（臂长为 $l=0.65\lambda$）

单位阶跃序列是离散系统中的一种序列类型，其定义为：$u(n-n_0)=\begin{cases}1, & n\geq n_0\\0, & n<n_0\end{cases}$

```
》n=-10：10；  n0=-2；
》u= (n>=n0)；
》whitebg ('w')；
》stem (n, u,'-.')；
》axis ( [-10 10 -.5 1.5])；
```

· fill 函数

fill 函数用于绘制二维多边形的填充图，其用法是：

》fill (x, y, c)　　用字符串 c 中确定的颜色，填充向量 x 和 y 定义的多边形。

例 14　家庭影院示意图

```
》fill ( [-5 5 5 -5], [0 0 2 2], [.9 .9 .9])；          % DVD
》axis off; axis equal;
》hold on;
》text (-.7, .9,'DVD')；
》fill ( [-4 4 4 -4], [0 0 6.5 6.5] +2.1, [.9 .9 .9])% 电视机
》fill ( [-1 -.75 -.75 -1] *3.333, [1 1 6 6] +2.01,'w')；
》fill ( [-.75 -.5 -.5 -.75] *3.333, [1 1 6 6] +2.01,'y')；
》fill ( [-.5 -.25 -.25 -.5] *3.333, [1 1 6 6] +2.01,'c')；
》fill ( [-.25 0 0 -.25] *3.333, [1 1 6 6] +2.01,'g')；
》fill ( [1 .75 .75 1] *3.333, [1 1 6 6] +2.01,'k')；
```

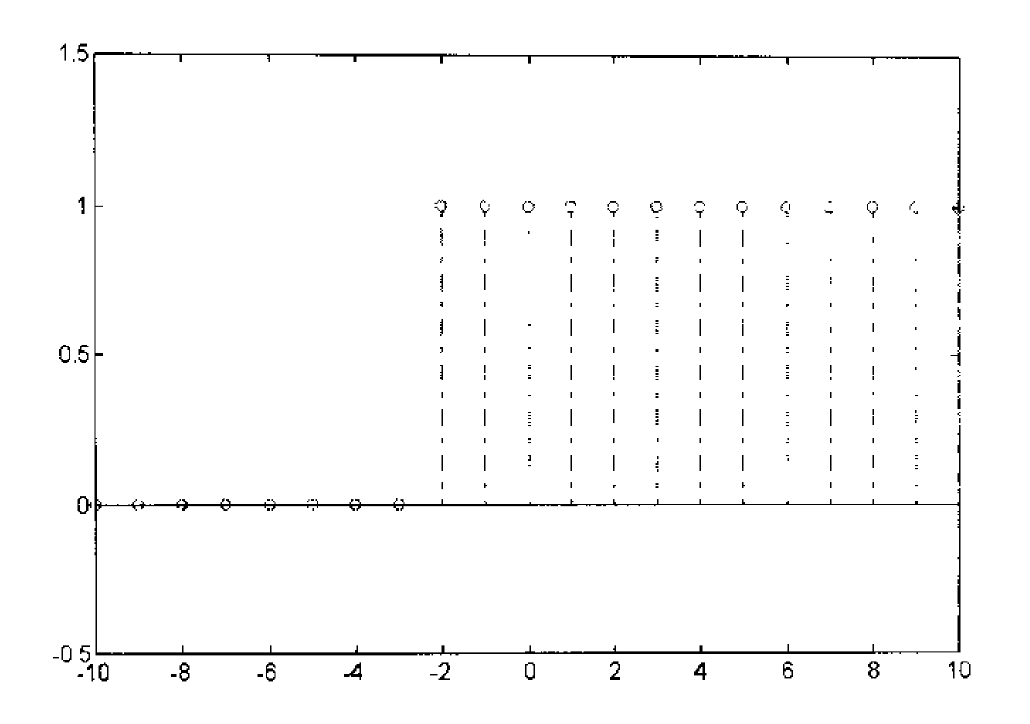


图 2.9 单位阶跃序列图

```
》fill ( [.75 .5 .5 .75] *3.333, [1 1 6 6] +2.01,'b');
》fill ( [.5 .25 .25 .5] *3.333, [1 1 6 6] +2.01,'r');
》fill ( [.25 0 0 .25] *3.333, [1 1 6 6] +2.01,'m');
》fill ( [6 10 10 6], [0 0 9 9], [.9 .9 .9]);          % 音箱
》fill ( [6.3 9.7 9.7 6.3], [.3 .3 8.7 8.7], [.7 .7 .7]);
》fill (- [6 10 10 6], [0 0 9 9], [.9 .9 .9]);         % 音箱
》fill (- [6.3 9.7 9.7 6.3], [.3 .3 8.7 8.7], [.7 .7 .7]);
》bold off;
```

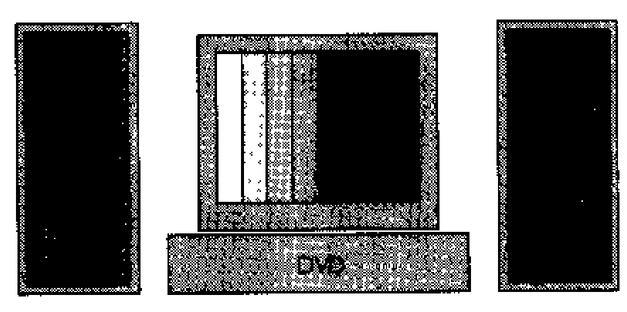


图 2.10 家庭影院示意图

绘制二维图形的函数还有很多，如：comet, bar, stairs, errorbar, hist, rose, compass, feather 等等，读者可以阅读专门介绍 MATLAB 的有关书籍，或使用联机帮助，在 4.2 版中，可以键入 help plotxy；而在 5.x 版中，则应键入 help graph2d，此处就不再赘述了。

2.6 M文件和M函数

对于简单的问题来说，在MATLAB的工作环境下直接输入命令是很方便的，如本章前几节给出的例子，而在处理比较复杂的问题过程中，最好的方法是使用M文件或M函数。

2.6.1 M文件

M文件又被称为脚本文件。从功能上来说，M文件类似于DOS系统中的批处理程序（*.BAT文件），它本身是个文本文件，可以用文本编辑器来建立并进行修改，其内容均为MATLAB的命令和函数。M文件的后缀为*.m。

M文件的使用方法是很简单的，设M文件的名称为abc.m，于是在MATLAB的工作环境下，直接键入abc，即可调用该M文件；在其它的M文件中写入abc字样，也可以调用该M文件。从这个意义上讲，M文件相当于使用者自己建立起来的一个MATLAB命令集。

M文件内的所有变量均为全程变量，即M文件中可以调用和改变当前工作空间内所有的变量。

在MATLAB命令窗口的File菜单下的New子菜单内选择M-file项，便可以启动文本编译器，进而建立一个M文件，而选择Open M-file项，则可以对已有的M文件进行编辑，见图1.4。

2.6.2 M函数

MATLAB中的M函数也是一种M文件，它功能和作用类似于FORTRAN语言中的子程序或C语言中的函数。M函数与M文件的区别主要在于：1）M函数的第一行一定是以function字样开头的；2）M函数中的变量为局部变量；3）M函数专门设有输入变量（参数）和输出变量（参数）。

一个M函数是用来完成一个特定的功能，即对其输入参数进行某种操作，以得到所需的输出参数。很多MATLAB本身的函数都是用M函数的方式写成的，因此编写若干M函数，使用者可以很方便地建立自己的工具箱来完成特定的工作任务。

使用MATLAB编程或仿真时，应该尽量采用M函数的方式。

·M函数的规范格式

M函数的规范格式可以分为四部分，1）函数名 2）函数说明 3）缺省参数的赋值 4）函数本身，具体形式如下：

```
function [a, b, c, d, e] = test-fun (x, y, z)
%
%    函数说明部分
%
%
```

缺省参数的赋值

函数本身

函数的第一行是以 function 字样开头的，说明这是一个 M 函数。第一行内还必须有此函数的函数名，在上面的例子中，函数名为 test _ fun，一般来说这个 M 文件的名称应该与函数名相同，即 test _ fun.m 。函数的名称应尽量遵守 8.3 规则。

函数名的前面为等号，等号左面为输出变量列表，各输出变量用中括号括起来，各变量之间用逗号分开；函数名的右面为输入变量列表，各输入变量用小括号括起来，各变量之间也用逗号分开。

函数可以不设输入变量和输出变量。

第二行是以 % 开头的，这表明其后为注释部分，此行又称为 H1 行，lookfor 命令就是对各函数的 H1 行进行检索的，因此可将此函数的基本功能在 H1 行内简单地表明，以便查询。

函数的说明部分从第二行开始，是由连续的注释行组成的，使用 help 命令就可以将函数的说明部分显示出来，这就是联机帮助的一种基本方式。在函数的说明部分，通常包括了函数的功能、函数的使用方法、函数的输入参数和输出参数、函数的作者、函数的修改记录等项内容。

根据变量 nargin，可以确定输入参数的数目，于是便可以对未输入的参数进行缺省值的赋值。

根据变量 nargout ，可以确定函数的调用过程中，输出了哪几个参数，在有些情况下，函数的具体功能可以根据输出参数的数量来进行调整。

例 15　一个典型的函数格式

```
function [a, b, c, d, e] =test _ fun (x, y, z, S)
%  Example of Function Format
%
%  Usage: function [a, b, c, d, e] = test-fun (x, y, z, S)
%
%  parameter x is , parameter y is
%  parameter z is , parameter S is
%  parameter a is , parameter b is , parameter c is
%  parameter d is , parameter e is
%
%  Simulation of Electronic System and MATLAB, Dec 1999

if  nargin<4; S='BBI';            end;
if  nargin<3; z=56+36*j;          end;
if  nargin<2; y=pi*3.4;           end;
if  nargin<1; x= (1: 100) *pi;    end;
```

```
    函数本身
```

· M 函数的使用规则

1）文件名应与函数名相同，这样比较清楚。

2）MATLAB 的运行过程中，第一次执行一个 M 函数时，先将文本文件打开，然后对函数进行编译，以加快运行速度；而对于 M 文件，MATLAB 是采用解释执行的方式，即解释一行运行一行。

3）M 函数有自己专有的工作空间，函数内的变量只是通过函数的输入参数和输出参数与 MATLAB 的工作空间联系在一起。

4）M 函数可以递归调用，即 M 函数可以自己调用自己。

5）M 函数的文件名（不包括它的后缀）就象一个 MATLAB 命令（函数）一样，在 MATLAB 的工作环境内键入函数的文件名，或在 M 文件及别的 M 函数中输入函数的文件名，就可以完成函数的调用。

6）在一个子目录之内，M 函数的名称是唯一的；而在不同的子目录内，可以有文件名相同的函数存在。

7）M 函数在调用过程中，一般来说其输入参数应该放在括号内，但是若输入参数为字符串时，则可以省略括号和引号，如 axis equal 与 axis（'equal'）是等价的，显然前者的形式比较简单。

·M 函数输入参数的其它方式

如上所述，M 函数的输入参数可以在调用的过程中进行确定，也可以采用缺省赋值的方式。在某些情况下，为了方便有时也可以采用全程变量；另外，在输入数据特别多时，还可以采用数据文件的方式。

全程变量用 global 函数来定义，如 global x y z m 就确定了四个全程变量。需要说明的是，全程变量应在函数调用之前就进行定义，这样全程变量在函数内就是有效的。由于函数内可以重新对全程变量进行赋值，因此在使用全程变量应该特别谨慎，在编程的过程中尽量避免使用全程变量。另外，全程变量的名称应该比较长一些，应该采用大写字母，这样做的目的是防止全程变量与局部变量混淆在一起。

数据文件是采用 load 命令调入的，具体做法见 2.7 节的有关部分。

· 子函数

在 5.x 版本的 MATLAB 中若要在一个 M 函数中调用另外一个使用者自己定义的函数(即子函数)，那么可以在该 M 函数的后面添加上子函数的程序段。子函数的格式与一般 M 函数相同，同样是以 function 字样开头的，也就是说将两个函数放置在一个 M 文件中了，这样减少了 M 文件的数目，同时函数之间的关系也比较明确。需要说明是，子函数只能被上述的 M 函数调用，而在其它的 M 函数中是不能调用它的。

· M函数的一个实例

为了使读者对M函数的编写和使用有一个比较全面的了解，此处给出一个M函数的实例。specanal函数是一个比较实用的仿真频谱分析仪和仿真示波器，可以用来对信号进行频谱分析。读者可以先将程序输入计算机，然后自己去仔细体会函数各部分的作用，这是学习MATLAB编程的一种好方法。该程序中使用的各种函数在本书中陆续介绍。

程序清单如下：

```
function [] =specanal (x, fs);
%
%   A simulated oscilloscope and a simulated spectrum analyzer
%
%   Usage: specanal (x, fs);          BBI  97
%   x - Signal Series,     fs - Sampling Frequency

if  nargin<2;    fh=15625;    fs=100 * fh;    end;          % 采样频率的缺省值
dt=1/fs;                                                     %采样时间间隔
if nargin<1;    t=0: dt: 40/fh - dt;    u=2 * pi * fh * t;
   x=sin (u)  + .1 * sin (2 * u)  + .01 * sin (3 * u)  + .316 * cos (4 * u);
end;        % 信号序列的缺省值:   电视的行频及其2次、3次和4次谐波
H1=figure ('Name','OSCILLOSCOPE     BBI 97/07', …
   'NumberTitle','off','Position', [10 120 560 420]);
                                                             %打开仿真示波器的窗口
uimenu ('Label','&Grid','Callback','grid;');                 % 建立grid菜单
N=length (x);    t=  (0: N-1)  * dt; T=N * dt;    ss='s';
if T<1e-03;    t=t * 1e+06;    ss='μs';          % 建立时间序列，并确定其单位
elseif T<1;    t=t * 1000;    ss='ms';
end;
whitebg ('k'); plot (t, x,'y');                              % 仿真示波器
set (gcf,'Color', [0 0 0]);
set (gca,'FontName','Arial','FontSize', 10);
xlabel ( ['t (' ss ')']); ylabel ('V');
zoom xon;    pause (1);            % x轴放大，并暂停1秒
% ==========================================
H2=figure ('Name','SPECTRUM ANALYZER     BBI 97', ……
   'NumberTitle','off','Position', [220 40 560 420]);
                                   %打开仿真频谱发析仪的窗口
mstr=' [u, v]  =ginput (1); v=round (v * 10)  * .1;';
mstr=  [mstr 'u=num2str (u);  v=num2str (v);'];
mstr=  [mstr 'nstr=''SPECTRUM ANALYZER     BBI 97'';'];
```

```
mstr= [mstr 'nstr= [nstr blanks (15) u '','' blanks (5) v '' (dBm) ''];'];
mstr= [mstr 'set (gcf,''Name'', nstr);'];
gstr='grd=get (gca,''Ygrid''); if strcmp (grd,''on''); grd=''off'';';
gstr= [gstr 'else; grd=''on''; end; set (gca,''Ygrid'', grd);'];
uimenu ('Label','&Marker','Callback', mstr);        % 建立 Marker 菜单
uimenu ('Label','&Y-grid','Callback', gstr);        % 建立 Y-grid 菜单
fmax=max (fs/2);            k=1;          hs='Hz';        % 确定频率的单位
if fmax>1e+06;              k=1e-06;      hs='MHz';
elseif fmax>1e+03;          k=1e-03;      hs='kHz';
end;
f1=k/ (dt*N);               % frequency discrimination (频率分辨率)
y=fft (x) /N;           y=abs (y) *2;           % FFT
N2=fix (N/2);           y=y (1: N2);
yn= (rand (size (y)) -0.5) *4;                  % 背景噪声仿真
yn=10.^ (yn*.05) *5e-05;
y=20*log10 (y+yn) +10;                           % 线性-对数转换 dBm, 50Ω
plot (f1* (0: N2-1), y,'g');                     % 仿真频谱分析仪
set (gcf,'Color', [0 0 0]);
set (gca,'Ygrid','on','FontName','Arial','FontSize', 10);
xlabel ( ['frequency (' hs ')']);
ystr=' (dBm)';     ylabel (ystr);
title ('Impedance: 50 Ohm, (dBμV) = (dBm) + 107 ');
zoom xon;
```

specanal 函数的两个输入参数为信号序列和采样频率，而其它的参数如：时间序列、采样时间间隔、频率分辨率、采样时间区间等，全可以由上述的两个输入参数求出来。缺省的输入信号序列是电视的行频（15625Hz），以及它的二次、三次和四次谐波成分，其相对强度分别为－20dB、－40dB 和－10dB；缺省的采样频率为行频的 100 倍频，利用函数的这些缺省值在编程的过程中可以对程序的正确性进行检验。

specanal 函数未设输出参数。

specanal 函数先打开仿真示波器的窗口，暂停 1 秒后，再打开仿真频谱分析仪的窗口，使用鼠标可以在两个窗口间进行切换。为了便于读数，在两个窗口内都设置了放大功能，使用鼠标的左键点击感兴趣的区域，即可对图形的横坐标进行放大，而纵坐标保持固定不变；使用鼠标的右键点击，则可恢复原样。在对频谱分析仪进行仿真的过程中，人为地添加上了强度为－76dBm 的背景噪声。图 2.11 为 specanal 函数使用缺省 参数时的显示内容。用鼠标点击 Grid 及 Y－grid 菜单，可给图形添加或去除栅格；点击 Marker 菜单则调用了 ginput 函数，以便使用鼠标来读取数据。关于菜单的使用，见第三章第五节。

例 14　电视行脉冲信号（行脉冲宽度为 4.7μs，周期为 64μs，采样时间间隔取 0.1μs）

```
》x= [  0.27*ones (1, 47)   0.67*ones (1, 640-47)]; x= [x x x x];
```

```
%四个周期
》dt=0.1e-06;         specanal (x, 1/dt);
```

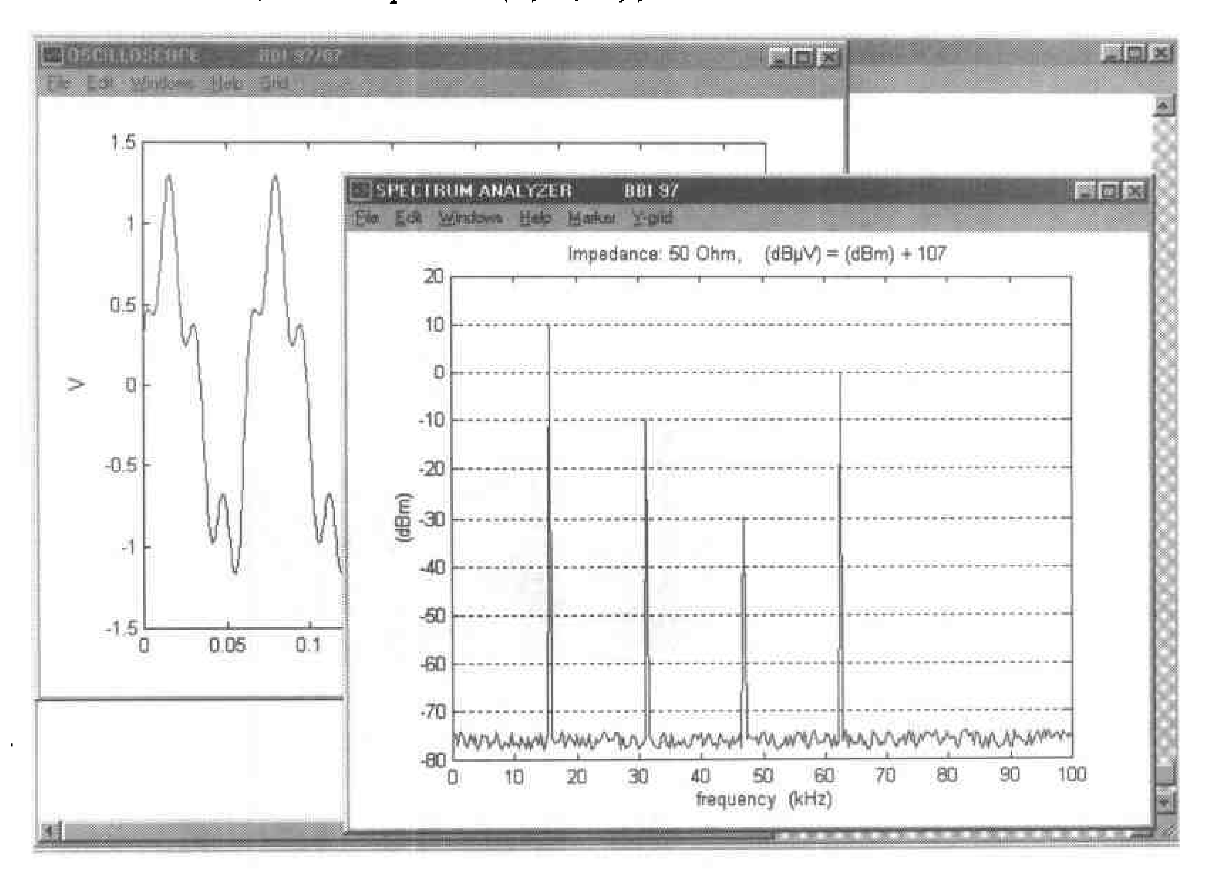


图2.11 仿真频谱分析仪与仿真示波器

2.7 数据文件与图形文件的输入和输出

在MATLAB中设立了load函数和save函数，对于大多数使用者来说，使用这两个函数对数据进行读取和储存是足够的了，而MATLAB还提供了低级的文件输入/输出函数集，可以用于读写任何已知的文件格式。

2.7.1 二进制格式和ASCII格式文件的输入和输出

· save函数

save函数将当前变量储存在文件中，二进制文件的后缀为mat；ASCII格式的文件后缀可由使用者指定，通常为dat或dta，此种格式的文件为文本文件，可以用文本编辑器将其打开。

save函数的用法是：

》save tmp x y;　　　　采用二进制格式，将变量x和y储存在tmp.mat文件中

》save tmp;　　　　采用二进制格式，将当前所有的变量储都存在tmp.mat文件中。

》save tmp.dat x y -ascii;　　　　采用ASCII码，将变量x和y储存在tmp.dat文件中，数据的长度为8位。

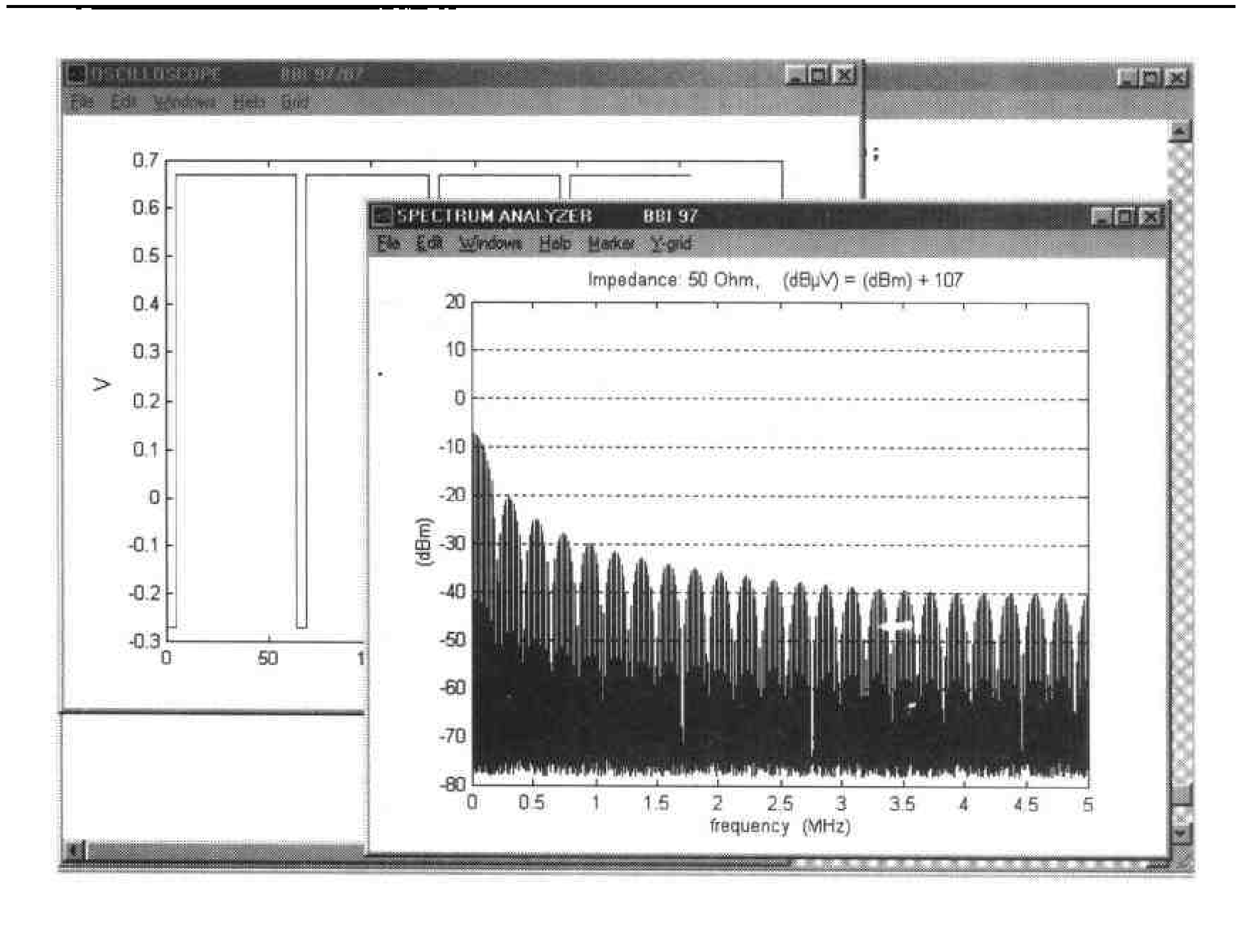


图 2.12 电视的行脉冲信号及其频谱

》save tmp.dat -ascii;　　采用 ASCII 码，将当前的所有变量储存在 tmp.dat 文件中，数据的长度为 8 位。

》save tmp.dat -ascii -double;　　采用 ASCII 码，将当前的所有变量储存在 tmp.dat 文件中，数据的长度为 16 位。

例 17　采用 ASCII 格式储存数据文件

》x=2:2:32;　　save tmp.dat x -ascii;

文本文件 tmp.dat 中的具体内容为：

```
2.0000000e+000   4.0000000e+000   6.0000000e+000   8.0000000e+000
1.0000000e+001   1.2000000e+001   1.4000000e+001   1.6000000e+001
1.8000000e+001   2.0000000e+001   2.2000000e+001   2.4000000e+001
2.6000000e+001   2.8000000e+001   3.0000000e+001   3.2000000e+001
```

· load 函数

load 函数用于读取 MAT 文件内储存的变量，或读取 ASCII 文件内储存的数据，其用法为：

》load tmp;　　读取 tmp.mat 文件内储存的全部变量。

》load tmp.dat;　　读取 tmp.dat 文件内储存的数据，并赋值给变量 tmp（即变量名与文本文件名相同）；另外，tmp.dat 文件内数据格式应该依据

m 行 n 列的方式储存的。

通常，使用 load 函数读取 MAT 格式的数据文件给 M 函数输入参数；而文本文件则往往用于连接 MATLAB 程序和用其它高级程序语言编写的程序，例如由于使用 C 语言或 FORTRAN 语言编写的程序将计算结果存储在文本文件中，然后使用 load 函数便可将该数据输入 MATLAB 的 M 函数中。MATLAB 的突出特点之一是具有功能强大的绘图功能且界面十分友好，而它的缺点则是运算速度较慢，故可采用数据文件的方式可以将 MATLAB 程序与 C 语言或 FOTRAN 语言联系在一起，以取长补短。

2.7.2 图形的拷贝

MATLAB 绘图函数绘制出来的各种图形和曲线可以通过图形窗口的 edit 菜单栏中的 copy 项（4.2 版）或 copy figure 项（5.x 版）拷贝到 Windows 系统中的剪贴板上，然后可以使用图形软件（Windows 中的 Paint、Photoshop 等）将图形转换成为 BMP、JPG、TIFF、GIF 等格式的图形文件储存起来。

使用键盘上的 print screen 键可以将当前屏幕上显示的内容拷贝到剪贴板上。

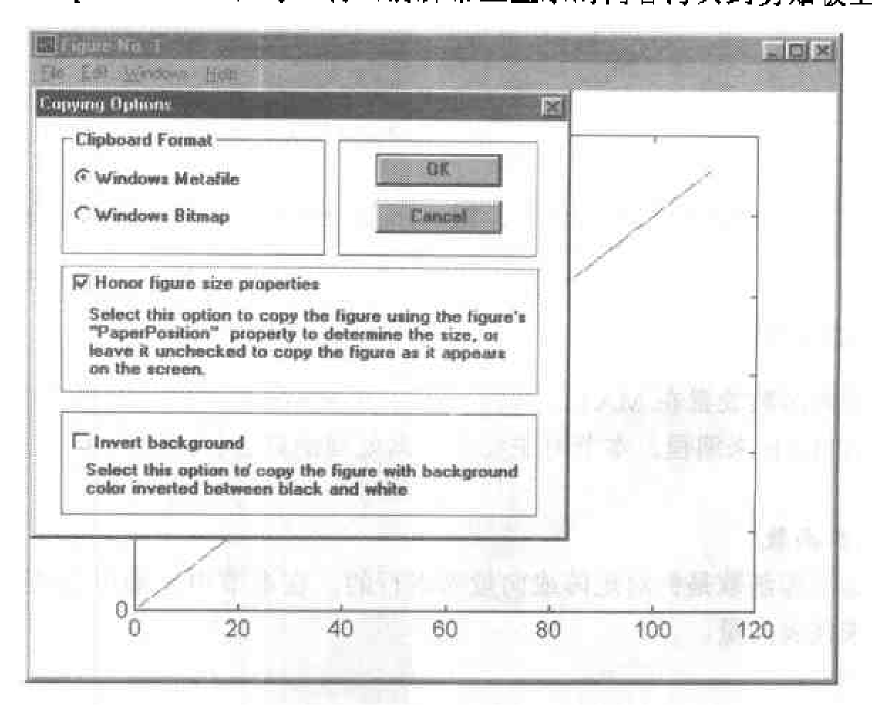


图 2.13 图形拷贝选项窗口（4.2 版）

在将图形拷贝到剪贴板的过程中，可以通过图形窗口的 edit 菜单栏中的 Copying Options 项（4.2 版）来选择拷贝文件的类型或对图形的背景颜色进行转换，如图 2.13 所示。一种文件类型称为图元文件（Metafile），另一种文件类型称为位图（Bitmap）。图元文件是矢量化的，便于进行放大和缩小，因此最好采用图元格式来拷贝图形。另外，若原图形的背景是黑色的，在拷贝过程中最好将它转换成为白色，因为这样便于进行打印。

在 5.x 版的 MATLAB 中，先在图形窗口的 File 菜单下的 Preferences 子菜单中找到 Copying Options 项，然后便可对拷贝文件的类型及图形的背景进行选择。

2.7.3 低级的文件输入和输出方式

除了 load 函数和 save 函数之外，MATLAB 还提供了基于 C 语言实现的低级文件的输入输出函数（I/O 函数），使用这些 I/O 函数可以对任意已知的文件格式进行读写，表 2.7 列出了一些主要的 I/O 函数。

表 2.7 低级 I/O 函数一览表

I/O 函数	功　能
fclose	关闭文件
feof	测试文件结束
ferror	查询文件 I/O 的错误状态
fgetl	读文件的行（忽略回行符）
fgets	读文件的行（包括回行符）
fopen	打开文件
fprintf	把格式化数据写入文件
fread	从文件中读取二进制数据
frewind	返回到文件开始
fscanf	从文件中读取格式化数据
fseek	设置文件位置指示符
ftell	获取文件位置指示符
fwrite	把二进制数据写入文件

读者可以在 MATLAB 的工作环境下，键入 help iofun 来查询各种低级的文件输入和输出函数的用法。

2.8 数据处理函数

各种数据处理函数放置在 MATLAB 的一个子目录 Datafun 内，了解这些函数的功能便于读者使用 MATLAB 来编程，本节对主要的数据处理函数进行简要的介绍。

2.8.1 基本函数

基本的数据处理函数是针对矩阵或向量来进行的。在本节中，采用 A 变量来代表矩阵，而采用 v 变量来代表向量。

·max 函数

max 函数用于查找矩阵或向量的最大值，并给出最大值的下标，其用法是：

》[vm, I] =max (v);　　vm 为最大值，它是一个数；而 I 则表示最大值的下标。

》[vm, I] =max (A);　　运算针对列向量进行，vm 为一个行向量，其中各元素为每列的最大值；I 也为行向量，其元素表示个最大值的下标。

例如：》[vm, I] =max ([1: 10　1: 10]);　　运算结果为 vm: 10　I: 10

》A= [1: 4; 2 6 2 3; 0 4 6 8; 5 3 4 1];　　[vm, I] =max (A);

运算结果为　　vm: [5 6 6 8]　　I: [4 2 3 3]

》A= [1: 4; 2 6 2 3; 0 4 6 8; 5 3 4 1];　　[vm, I] =max (A (:));

先将矩阵按列转换成为向量，然后求向量的最大值，其结果为vm：8 I：15

·min函数

min函数用于查找矩阵或向量的最小值，并给出其下标，其用法与max函数类似：

》[vm，I] =min（v）； vm为最小值，它是一个数；而I则表示最大值的下标。

》[vm，I] =min（A）； 运算针对列向量进行，vm为一个行向量，其中各元素为每列的最小值；I也为行向量，其元素表示个最大值的下标。

·mean函数

mean函数用于求向量和矩阵的平均值，其用法是：

》vm=mean（v）； vm为各元素的平均值，它是一个数。

》vm=mean（A）； 运算针对列向量进行，vm为一个行向量，其中各元素为矩阵中每列元素的平均值。

·median函数

median函数用于求向量和矩阵的中值，其用法是：

》vm=median（v）； vm为各元素的中值，它是一个数。

》vm=median（A）； 运算针对列向量进行，vm为一个行向量，其中各元素为矩阵中每列元素的中值。

请注意中值的含义，当向量有偶数个元素时中值是取数值上两个相邻元素的平均值。如：

》median（[1 2 0 5]） 运算结果为1.5，即表示有两个元素大于1.5，有两个元素小于1.5，两相邻元素是1和2。

》median（[3 -5 8 1]） 运算结果为2，即表示有两个元素大于2，有两个元素小于2，两相邻元素是1和3。

· std函数

std函数用于求向量和矩阵的标准差（又称为标准偏差、标准误差或均方差），其计算公式为：$s=\sqrt{\frac{\sum x_i^2 - n\cdot \bar{x}^2}{n-1}}$。std函数的用法是：

》s=std（v）； s为各元素的标准差，它是一个数。

》s=std（A）； 运算针对列向量进行，s为一个行向量，其中各元素为矩阵中每列元素的标准差。

· sort函数

sort函数用于对向量元素和矩阵元素进行排序(从小到大)，并给出下标向量和下标矩阵，其用法是：

》[vs，I] =sort（v）； vs排序后的向量，I为下标向量，实际上vs=v（I)。

》[As, I] =sort (A);　　As 排序后的矩阵（按列向量进行排序），I 为下标矩阵。

》A= [1: 4; 2 6 2 3; 0 4 6 8; 5 3 4 1]; [As, I] =sort (A);　　运算结果为:

$$A=\begin{matrix}1&2&3&4\\2&6&2&3\\0&4&6&8\\5&3&4&1\end{matrix}\quad A_s=\begin{matrix}0&2&2&1\\1&3&3&3\\2&4&4&4\\5&6&6&8\end{matrix}\quad I=\begin{matrix}3&1&2&4\\1&4&1&2\\2&3&4&1\\4&2&3&3\end{matrix}$$

· sum 函数

sum 函数用于对向量元素求和，或对矩阵中列向量元素进行求和，其用法是:

》s=sum (v);　　s 为向量中各元素的和，它是一个数。

》s=sum (A);　　s 为一个行向量，其中各元素为矩阵中每列元素的和。

· prod 函数

prod 函数用于对向量元素求积，或对矩阵中列向量元素进行求积，其用法是:

》p=prod (v);　　p 为向量中各元素的积，它是一个数。

》p=prod (A);　　p 为一个行向量，其中各元素为矩阵中每列元素的积。

· cumsum 函数

cumsum 函数用于对向量元素求累加和，或对矩阵中列向量元素进行求累加和，其用法是:

》s=cumsum (v);　　s 为向量中各元素的累加和，它是一个数。

》s=cumsum (A);　　s 为一个行向量，其中各元素为矩阵中每列元素的累加和。

· cumprod 函数

cumprod 函数用于对向量元素求累乘积，或对矩阵中列向量元素进行求累乘积，其用法是:

》p=cumprod (v);　　p 为向量中各元素的累乘积，它是一个数。

》p=cumprod (A);　　p 为一个行向量，其中各元素为矩阵中每列元素的累乘积。

2.8.2 有限差分

· diff 函数

diff 函数的功能是差分，其定义是：已知 v 为一个向量，它共有 n 个元素，

设 u=diff (v)，于是 $u_i=v_{i+1}-v_i$，i=1, 2, …n−1

对于矩阵来说，运算是针对列向量进行的。

》u=diff (v);　　u 为一个向量，其中的元素数量为 length (v) −1 个。

》B=diff (A);　　B 为一个矩阵，若 A 为 m×n 的矩阵，则 B 为 (m−1) ×n 的矩阵。

》A= [1: 4; 2 6 2 3; 0 4 6 8; 5 3 4 1]; B=diff (A); 运算结果为:

$$A=\begin{matrix}1&2&3&4\\2&6&2&3\\0&4&6&8\\5&3&4&1\end{matrix}\qquad B=\begin{matrix}1&4&-1&-1\\-2&-2&4&5\\5&-1&-2&-7\end{matrix}$$

例 18　方波信号的近似微分

使用 diff 函数可以找到向量发生突变的地方，并可近似计算微分。

```
》v= [ones (1, 10) zeros (1, 10) ones (1, 10)];
》v= [v v v];      n=1: length (v);
》u= [0 diff (v)];    whitehg ('w');
》plot (n, v,'b', n, u-1.2,'r');    axis off;
》text (95, 1,'v');    text (95, -1.2,'u');
```

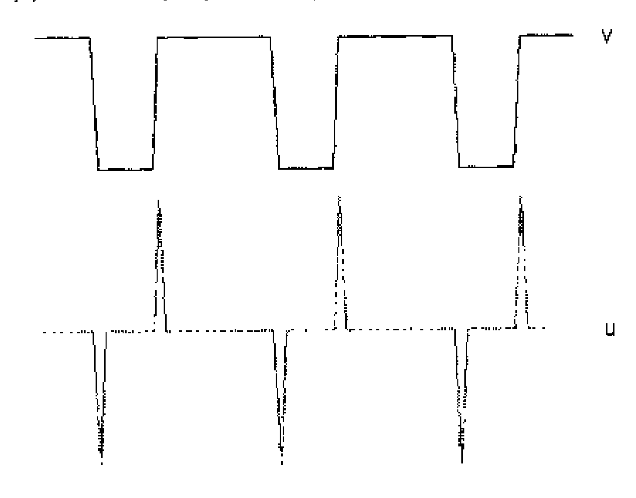


图 2.14　方波信号的近似微分

· gradient 函数

gradient 函数的功能是用来近似求梯度。[px, py] =gradient (Z, dx, dy) 返回矩阵 Z 的数值偏微分，其中 px=dZ/dx, py=dZ/dy；而 [px, py] =gradient (Z) 则表示 px=py =1。

2.8.3　矢量运算

矢量本身就是向量，因此矢量运算在 MATLAB 中是十分方便的。

·dot 函数

dot 函数的功能是做点乘。C=dot (A, B) 返回的是矢量 A 和 B 的点积（数量积），即 $C=\boldsymbol{A}\cdot\boldsymbol{B}$

· cross 函数

cross 函数的功能是做叉乘。C=cross (A, B) 返回的是矢量 A 和 B 的叉积（矢量积），即 $\boldsymbol{C}=\boldsymbol{A}\times\boldsymbol{B}$

```
》A= [1 2 4]; B= [4 3 2]; C= [1 1 2];
》D=dot (A, cross (B, C));          三重标积，其结果为-4
》E=cross (cross (A, B), C);        二重矢积，其结果为 [33   11   -22]
```

2.8.4 相关

根据信号分析理论，随机变量可以用其分布函数、概率密度和数字特征来描述，而随机变量的数字特征主要有方差、协方差、相关系数等。

· cov函数

cov函数用来计算方差和协方差，其计算公式为：$b_{12}=\dfrac{\sum_i (x_i-\bar{x})(y_i-\bar{y})}{n-1}$。cov函数用法是：

》d=cov (v);　　v为一个向量，d是该向量的方差。

》B=cov (u, v);　　u和v都是向量，B为一个2×2的协方差矩阵，其中B11和B22为方差，B12和B21为协方差。

》B=cov ([u v]);　　当u和v都是列向量时，cov ([u v]) 与cov (u, v) 等价。

》B=cov (A);　　A为一个m×n矩阵时，则B为一个n×n的协方差矩阵。具体运算是按A矩阵的列向量进行的，其中A矩阵的列向量为变量，行向量为观察行。

·corrcoef函数

corrcoef函数用来计算相关系数，根据数理统计的理论，相关系数可以由协方差求出：$r_{12}=\dfrac{b_{12}}{\sqrt{b_{11}\cdot b_{12}}}$。corrcoef函数的用法是：

》R=corrcoef (u, v);　　u和v都是向量，R为一个2×2的相关系数矩阵。

》R=corrcoef ([u v]);　　当u和v都是列向量时，corrcoef ([u v]) 与corrcoef (u, v) 等价。

》R=cov (A);　　A为一个m×n矩阵时，则B为一个n×n的相关系数矩阵。具体运算是按A矩阵的列向量进行的，其中A矩阵的列向量为变量，行向量为观察行。

2.8.5 滤波与卷积

· filter函数

filter函数的作用是对一维的离散信号进行数字滤波。数字滤波器通常是用两组系数来描述的，其中一组系数为a，其中有m个元素，而另一组系数为b，其中有k个元素。数字滤波是用差分来完成的，设输入信号向量为x，数字滤波器输出的信号向量为y，于是有：

$$y_n = b_1x_n + b_2x_{n-1} + b_kx_{n-k+1} - a_2y_{n-1} - \cdots - a_my_{n-m+1}$$

filter函数的用法是：

》y=filter (b, a, x);　　x为输入向量，y是输出向量，b和a为滤波器的系数。

· conv 函数

conv 函数的作用是计算两序列（向量）间的卷积或计算多项式乘法，其计算公式为：$x(n) * y(n) = \sum_k x(k) \cdot y(n-k)$。conv 函数的用法是：

》z = conv (x, y);　　　z 为两序列 x 和 y 的卷积。

例 19　计算 $(x^2+2x-1) \cdot (x^2+1)$

两个多项式的系数向量分别为 [1 2 -1] 和 [1 0 1]，多项式相乘实际上就是系数向量的卷积。

》a = conv ([1 2 -1], [1 0 1])

计算结果为 [1 2 0 2 -1]，这表示 $(x^2+2x-1) \cdot (x^2+1) = x^4+2x^3+2x-1$

· deconv 函数

deconv 函数的作用是计算两序列（向量）间的逆卷积或计算多项式除法，其用法是：

》[q, r] = deconv (x, y);　　　q 为逆卷积的计算结果，而 r 为残数向量，卷积与逆卷 积之间的关系为：y = conv (q, x) + r，对于多项式除法来说，q 为商，而 r 为余数。

例 20　多项式除法

要计算 $(x^2+2x+3) / (x+1)$，于是

》[q, r] = deconv ([1 2 3], [1 1])

计算结果为 q = [1 1]，r = [0 0 2]，这表示 $(x^2+2x-1) / (x+1) = x+1 \cdots\cdots 2$

2.8.6　离散傅立叶变换

离散傅立叶变换是信号分析中的一种重要工具，它将时域内的问题转化成为频域内的问题，在很多情况下可以大大地简化问题的求解过程，另外计算时域信号的频谱也主要依靠离散傅立叶变换来完成，因此它在电子系统的仿真过程起着极其重要的作用。

· fft 函数

fft 函数的作用是对一序列（向量）作离散傅立叶变换，其计算公式为：

$$X_k = \sum_{n=1}^{N} x_n \cdot \exp[-j2(k-1)(n-1)\pi/N], \quad 其中\ k = 1, \cdots, N。$$

需要说明的是，由于 MATLAB 中向量的下标不允许出现零，因此与通常的离散傅立叶变换公式相比，上式的下标移动了一位。fft 函数的用法是：

》y = fft (x);　　　y 是向量 x 的离散傅立叶变换。

· 离散傅立叶变换与傅立叶级数之间的关系

傅立叶级数是计算周期函数频谱的依据，对于一个满足收敛条件的周期函数 f (t)，其傅立叶级数的表达式为：$f(t) = \sum_{n=-M}^{N} c_n \exp(jn\omega_1 t)$，中 $\omega_1 = \frac{2\pi}{T} = 2\pi f_1$

c_n 称为傅立叶级数的系数，它代表了频率为 nf_1 的频率分量的强度，即频谱强度。c_n 的

计算公式为：

$$c_n = \frac{1}{T}\int_0^T f(t)\exp(-jn\omega_1 t)dt = \frac{1}{N}\sum_{k=0}^{N-1} f(kT/N)\exp(-j2kn\pi/N)$$

对比 FFT 和傅立叶系数的计算公式，可以看出只需将上式中的 k 改成 k-1，将 n 改成 n-1，傅立叶级数的系数可以由离散傅立叶变换求出，即

$$c_n = \frac{1}{N}\sum_{k=1}^{N} f(k\triangle t)\exp[-j2(k-1)(n-1)\pi/N]，其中 \triangle t = T/N。$$

一般来说，傅立叶级数的系数为复数，设：$c_n = a_n + jb_n$，则周期函数的余弦级数和正弦级数可以表示为：

$$f(t) = a_0 + \sum_n (2a_n \cdot \cos n\omega_1 t - 2b_n \cdot \sin n\omega_1 t)$$

上式将傅立叶级数与离散傅立叶变换联系在一起了。

对于离散傅立叶变换 y=fft (x) 来说，y (1) 是信号 x 中的直流分量，2 * abs (y (2)) 是频率为 f_1 的频率分量的强度，2 * abs (y (3)) 是频率为 $2f_1$ 的频率分量的强度，余此类推；而 $f_1 = 1/T$，可称为频率分辨率。

例 21　正弦信号与余弦信号的离散傅立叶变换

```
》fs=100; dt=1/fs; T=0.1; t=0: dt: T-dt;
》x=2*cos (2*pi*10*t) +cos (2*pi*20*t);
》x=x+sin (2*pi*10*t) +.5*sin (2*pi*40*t);
》y=fft (x) /length (x);
》c=real (y) *2; s=-imag (y) *2;
》whitebg ('w'); subplot (211); stem (c);
》title ('Cosine Series');
》subplot (212); stem (s);
》title ('Sine Series');
```

由于 $f_1 = 1/T = 10$，因此离散傅立叶变换的计算结果与输入的信号是完全吻合的。

· fftshift 函数

从图 2.15 可以清楚地看出，当 N 为偶数时（为通常的情况），离散傅立叶变换的计算结果的对称轴位于$\frac{N}{2}+1$处（直流成分除外），对称轴上面的部分代表了所谓的负频率项；而 fftshift 函数的功能就是平移零点到频谱的中心，其用法是：

```
》y=ffitshift (y);
```

在例 19 中的程序后面再加上以下的程序段，从计算结果中可清楚地看出 fftshift 函数的作用。

```
》yy=fftshift (y); cc=real (yy) *2; ss=-imag (yy) *2;
》[c; cc; s; ss]
c :   [0.0  2.0  1.0  0.0  0.0  0.0  0.0  0.0  1.0  2.0]
cc:   [0.0  0.0  0.0  1.0  2.0  0.0  2.0  1.0  0.0  0.0]
s:    [0.0  1.0  0.0  0.0  0.5  0.0  -0.5  0.0  0.0  -1.0]
```

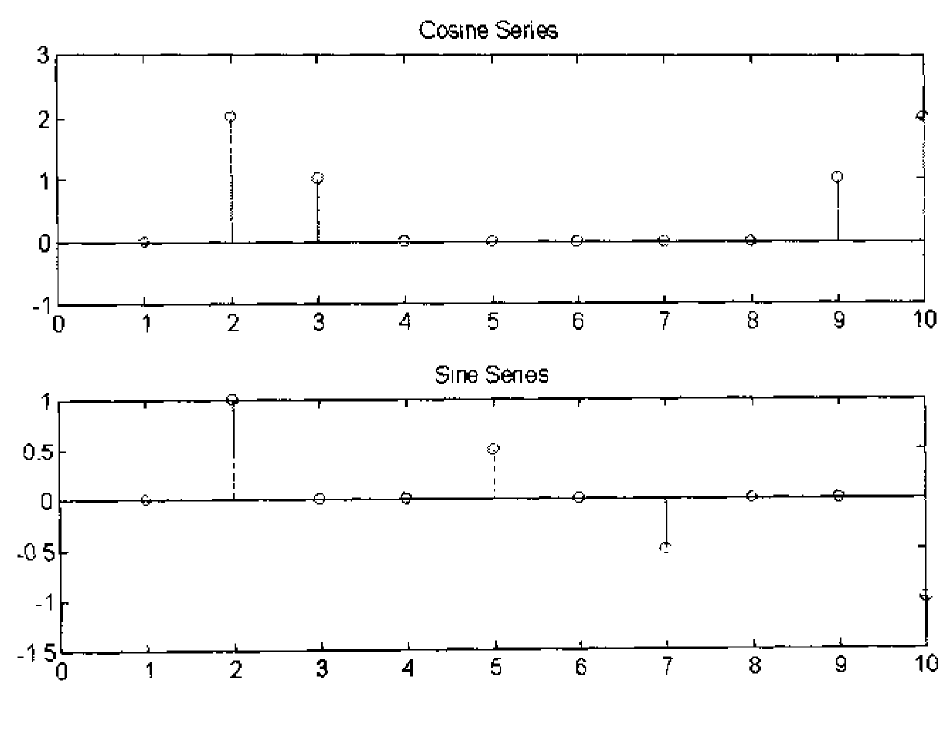


图 2.15　正弦和余弦级数的系数

ss:　[0.0　−0.5　0.0　0.0　−1.0　0.0　1.0　0.0　0.0　0.5]

· ifft 函数

ifft 函数的作用中进行离散逆傅立叶变换，其用法是：

〉x = ifft（y）;　　　　y 是向量 x 的离散傅立叶变换。

·nextpow2 函数

nextpow2（x）函数的作用是求出大于输入参数 x 绝对值的最小的 2 的幂次。

〉p = nextpow2（77）;　　　　计算结果为 p：7

2.9　字符串函数

各种与字符串有关的函数放置在 MATLAB 的子目录 strfun 内，MATLAB 提供了很强的字符串处理功能。在本节中，采用 s、s1、s2 等变量代表字符向量或字符矩阵。

2.9.1　一般函数

· abs 函数

abs 函数作用于字符串时，是求它对应的 ASCII 码，其用法是：

〉v = abs（s）;　　　　v 为行向量，其内的元素为字符的 ASCII 码。

〉v = abs（'abcdefghijklmnopqrstuvwxyz0123456789'）

计算结果为：v =

Columns 1 through 12

97　98　99　100　101　102　103　104　105　106　107　108

Columns 13 through 24

109 110 111 112 113 114 115 116 117 118 119 120

Columns 25 through 36

121 122 48 49 50 51 52 53 54 55 56 57

》 v=abs（'ABCDEFGHIJKLMNOPQRSTUVWXYZ'）

计算结果为：v =

Columns 1 through 12

65 66 67 68 69 70 71 72 73 74 75 76

Columns 13 through 24

77 78 79 80 81 82 83 84 85 86 87 88

Columns 25 through 26

89 90

·setstr 函数

setstr 函数的作用是将 ASCII 码向量转换成为对应的字符，其用法是：

》 s=setstr（a）；　　a 为向量，其数值范围是 0－255，s 是对应的字符串。

》 s=setstr（［66 66 73］）　　返回的计算结果为 s：BBI

· isstr 函数

isstr 函数用来判断一个变量是否为字符串，其用法是：

》 isstr（a）；　　若变量 a 为字符串或字符矩阵时，输出为 1；否则输出为 0。

· blanks 函数

blanks 函数用于定义空格符。

》 s=blanks（10）；　　s 为字符串，其内容为 10 个空格。

· dehlank 函数

deblank（s）的作用是去除字符串 s 尾部的空格（如果存在空格的话）。

·str2mat 函数

str2mat 函数的作用是将独立的字符串转换成为字符矩阵，为了确保每一行的长度相同，函数自动在较短的字符串的尾部添加空格。str2mat 函数执行一次，最多可将 10 字符串转换成字符矩阵，其用法是：

》 s=str2mat（s1，s2，s3，s4，s5，s6，s7，s8，s9，s10）；s 为 10 行的字符矩阵

超过了 10 行以后，可采用以下的方法形成字符矩阵

》 s=str2mat（s，s11，s12，s13，s14，s15，s16）；　　s 为 16 行的字符矩阵

· eval 函数

eval（s）函数的作用是执行字符串 s 内所包含的 MATLAB 命令、函数和程序段，此函

数常用在菜单界面和图形界面的环境下，它给使用者提供了很多便利，其用法是：

》eval (s)；　　执行字符串s内的MATLAB程序段，注意s不能是字符矩阵。

》[x, y, z, a] =eval (s)；　　执行字符串s内的MATLAB程序段，并返回输出变量。

2.9.2 字符串比较

· strcmp函数

strcmp函数的作用是对两个字符串进行比较，其用法是：

》strcmp (s1, s2)；　　若两字符串s1和s2相同，则输出为1；否则输出为0。

· findstr函数

findstr函数的作用是在一字符串中找到给定的字符串，其用法是：

》v=findstr (s1, s2)；　　在较长的字符串中寻找较短的字符串，若找到后则输出其出现的位置（下标）。

· upper函数

upper (s) 将字符串s内的小写字母均转换成为大写字母。

· lower函数

lower (s) 将字符串s内的大写字母均转换成为小写字母。

· isletter函数

isletter (s) 用于判断字符串s中的字符是否为字母，是字母时输出为1，否则为0。

如：》v=isletter ('Word 7.0')；　　输出为v：[1 1 1 1 0 0 0 0]

· strrep函数

strrep函数的功能是作字符串替换，其用法是：

》s=strrep (s1, s2, s3)；　　在字符串s1中，用字符串s3替换字符串s2。

2.9.3 转换函数

· num2str函数

num2str函数的功能是将数字向量或矩阵转换成为字符串或字符串阵，如：

》s=num2str (1.2)；　　输出结果s：'1.2'

· int2str函数

int2str函数的功能是将整数转换成为字符串，同时它对小数有自动取整的作用，如：

》s=int2str ([3.4 7 3.6])；　　输出结果s：'374'

· str2num函数

str2num 函数的功能是将数字字符串转换成为数字，如：

》 v=str2num（'4.2'）；　　　　　输出结果 v：4.2000

· mat2str 函数

mat2str 函数的功能是将数量矩阵转换为字符矩阵，如：

》 A=［1：3；4：6；7：9］；s=mat2str（A）；　输出结果 s：'［1 2 3；4 5 6；7 8 9］'

· hex2dec 函数

hex2dec 函数的功能是将十六进制数转换成为十进制数，注意在 MATLAB 中，十六进制数实际上是字符串。如：

》 v=hex2dec（'1AF'）　　　　　输出结果 v：431

· dec2hex 函数

dec2hex 函数的功能是将十进制数转换成为十六进制数，如：

》 s=dec2hex（255）　　　　　输出结果 s：'FF'

2.9.4　应用实例

为了使读者对字符串函数的使用有比较明确的了解，此处给出一个用字符串函数编写的实用函数 rom2arab，其功能是将罗马数字转换成为阿拉伯数字。

罗马数字共有Ⅰ，Ⅴ，Ⅹ，L，C，D，M 七个，依次表示以下数值 1，5，10，50，100，500，1000；另外，两个相同的数字并列表示相加，而两个不同的数字并列时，遵从'左减右加'的规则，即小的数字在左而表示相减，小的数字在右而表示相加。如：

Ⅰ	1	Ⅹ	10	LⅩⅩ	70	DCC	700
Ⅱ	2	Ⅺ	11	LⅩⅩⅩ	80	DCCC	800
Ⅲ	3	ⅩⅤ	15	ⅩC	90	CM	900
Ⅳ	4	XIX	19	C	100	M	1000
Ⅴ	5	ⅩⅩ	20	CC	200	MC	1100
Ⅵ	6	ⅩⅩⅩ	30	CCC	300	MD	1500
Ⅶ	7	ⅩL	40	CD	400	MDC	1600
Ⅷ	8	L	50	D	500	MCM	1900
Ⅸ	9	LⅩ	60	DC	600	MM	2000

函数 rom2arab 的程序清单如下：

```
function N=rom2arab (roma)
%     Roman to Arabic Number Conversion.
%
%      USAGE: N=rom2arab (roma)
%
%     BBI   1999
if nargin<1;    roma='MDCCCLIII';    end;          % 缺省输入
```

```
Tab=zeros (size (1, 22));        % 建立一个罗马字母与阿拉伯数字的对照表
Tab (1) =100;   Tab (2) =500;   Tab (7) =1;   Tab (10) =50;   % C D I L
Tab (11) =1000;   Tab (20) =5;   Tab (22) =10;                % MVX
a=abs (roma);                   % 求罗马数字所对应的 ASCII 码
I=find (a>=99);
if length (I) >0; a (I) =a (I) -32;   end;   % 转换为大写字母的 ASCII 码
a=a-66;   n=length (roma);   % 将 a 向量的数值范围限定为 [1, 22]
for i=1: n;
      b (i) =Tab (a (i));       % 查表，找到罗马字母对应的阿拉伯数字
end;
for i=1: n-1;
  if b (i) <b (i+1);   b (i) =-b (i);   end;   % 实施左减的规则
end;
N=sum (b);                      % 求和，得到结果
```

例如：键入 n=rom2arab ('MDCCCXII')，输出结果 n：1812

2.10 数学函数

MATLAB给使用者提供丰富的数学工具，这些数学函数分别放置在 MATLAB 的子目录 elfun，specfun 和 funfun 内。

2.10.1 基本数学函数

基本数学函数包括：三角函数、双曲函数、指数函数、对数函数、平方根函数、复数函数、取整函数、取余函数和符号函数。

三角函数中有正弦函数、反正弦函数、余弦函数、反余弦函数、正切函数、反正切函数、正割函数、反正割函数、余割函数、反余割函数、余切函数、反余切函数，这些函数的名称与通常的书写方式相同，但是采用弧度单位。

双曲函数的情况与三角函数类似。

exp (x) 为指数函数，log (x) 为自然对数函数，log10 (x) 为常用对数函数，sqrt (x) 为平方根函数。

· 取整函数

在 MATLAB 中共设置了四个取整函数，其中 fix (x) 向零方向取整（绝对值最小），floor (x) 向负无穷方向取整，ceil (x) 向正无穷方向取整，而 round (x) 则采用四舍五入的方式取整。

```
》v= [-2.6 -2.5 -2.4 4.2 4.5 4.8];
》n= [fix (v); floor (v); ceil (v); round (v)];
```

$$n = \begin{matrix} -2 & -2 & -2 & 4 & 4 & 4 \\ -3 & -3 & -3 & 4 & 4 & 4 \\ -2 & -2 & -2 & 5 & 5 & 5 \\ -3 & -3 & -2 & 4 & 5 & 5 \end{matrix}$$

·rem 函数

rem 函数的作用是取余数，其用法是：

》r=rem（n，m）；　　　　其中 n 是被除数，m 是除数，r 为余数。

》r=rem（1：10，4）；　　　计算结果为 r：[1 2 3 0 1 2 3 0 1 2]

·sign 函数

sign（x）称为符号余数。当 x>0 时，符号函数输出为 1，x = 0 时，输出为 0，x < 0 时，输出为 -1。

》v=sign（-2：5）；　　　计算结果为 v：[-1 -1 0 1 1 1 1 1]

2.10.2　特殊函数

在 MATLAB 中提供了很多特殊函数，如贝塞尔函数、勒让德函数、椭圆函数、贝塔函数、伽玛函数、误差函数等。

·贝塞尔函数

besselj（n，x）为第一类贝塞尔函数，通常的表达形式为 Jn（x），它是 n 阶贝塞尔方程 $\frac{d^2y}{dx^2}+\frac{1}{x}\cdot\frac{dy}{dx}+\left(1-\frac{n^2}{x^2}\right)y=0$ 的解，

bessely（n，x）为第二类贝塞尔函数，通常的表达形式为 $Y_n(x)$，又称为诺埃曼函数，第二贝塞尔函数可以用第一类贝塞尔函数表示出来 $Y_n(x)=\frac{\cos n\pi\cdot J_n(x)-J_{-n}(x)}{\sin n\pi}$。

besseli（n，x）为第一类变形贝塞尔函数，通常的表达形式为 $I_n(x)$，它又称为虚宗量贝塞尔函数，$I_n(x)=i^{-n}J_n(ix)$。

besselk（n，x）为第二类变形贝塞尔函数，通常的表达形式为 Kn（x），它又称为虚宗量汉克函数，$K_n(x)=\frac{\pi}{2}i^{-n}[J_n(ix)+Y_n(ix)]$。

·legendre 函数

legendre（n，x）称为连带勒让德函数，通常的表达形式为 $P_n^m(x)$，它是 m 阶连带勒让德方程 $\frac{d}{dx}\left[(1-x^2)\frac{dy}{dx}\right]+\left[n(n+1)-\frac{m^2}{1-x^2}\right]\cdot y=0$ 的解。

》P=legendre（n，x）；　　x 为一个数时，输出为一列向量 $[P_n^0(x)\ P_n^1(x)\cdots P_n^n(x)]'$，当 x 为一向量时，输出为一个 n 行的矩阵。

·beta 函数

beta (z, w) 称为贝塔函数 (又称为第一类欧拉积分), 通常的表达形式为 B (z, w), 其定义是: $B(z,w)=\int_0^1 t^{z-1}\cdot(1-t)^{w-1}\cdot dt$。

· gamma 函数

gamma (x) 称为伽玛函数 (又称为第二类欧拉积分), 通常的表达形式为 Γ (x), 其定义是: $\Gamma(x)=\int_0^{\infty}e^{-t}t^{x-1}dt$。

· erf 函数

erf (x) 称为误差函数, 其定义是: erf (x) $=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$

· 雅可比椭圆函数

雅可比椭圆函数包括: sn u (椭圆正弦)、cn u (椭圆余弦)、tn u (椭圆正切) 和 dn u。

设 $u=\int_0^x \frac{dx}{\sqrt{(1-x^2)(1-k^2x^2)}}$, 其中 k 称为模数, $0\leqslant k^2<1$, 此积分称为第一类椭圆积分。将 z 表示成 u 的函数 (即第一类椭圆积分的反函数), 记为 z = sn u, 这样就定义了椭圆正弦。

而 $\sqrt{1-z^2}=\sqrt{1-sn^2u}=cn\ u$ 、tn u = sn u/cn u 和 $\sqrt{1-k^2z^2}=dn\ u$ 则定义了另外几个雅可比椭圆函数。

MATLAB 表达式 [sn, cn, dn] = ellipj (u, k2) 同时给出了三种椭圆函数的值, 其中椭圆函数的自变量为 u (u 可以为向量), k2 代表模数 k 的平方。

· 完全椭圆积分

第一类完全椭圆积分的定义为: K (k) $=\int_0^1\frac{dx}{\sqrt{(1-x^2)(1-k^2x^2)}}$。第二类完全椭圆积分的定义为: E (k) $=\int_0^1\frac{\sqrt{1-k^2x^2}}{\sqrt{1-x^2}}dx$

MATLAB 表达式 [K, E] = ellipke (k2) 同时给出了两类完全椭圆积分的值, 其中输入参数 k2 代表模数 k 的平方, $0\leqslant k^2<1$。

·指数积分

指数积分的定义为: E_1 (x) $=\int_x^{\infty}\frac{\exp(-t)}{t}dt$, MATLAB 表达式 expint (x) 给出了指数积分的值。

· gcd 函数

gcd 函数的功能是求最大公约数, 其用法是:

》 g = gcd (a, b);　　　　为整数 a 和 b 的最大公约数。

· lcm 函数

lcm 函数的功能是求最小公倍数，其用法是：

》c=lcm (a, b);　　　　c 为正整数 a 和 b 的最小公倍数。

2.10.3 数值分析

· fplot 函数

fplot 函数的功能是进行函数绘图，与 plot 函数类似，它可以与 grid、zoom、title、xlabel、ylabel、hold 等函数一起使用，其用法是：

》fplot (fname, [xmin xmax]);　　输入参数 fname 为字符串，它可以是 M 函数的名称，也可以是 MATLAB 中承认的其它数学函数或数学函数的组合，自变量的名称应该为 x；输入的另一个参数为一个向量，它表示了函数自变量的变化区间。

》fplot (fname, [xmin xmax ymin ymax]);　　输入的第二个参数表示了函数变量的变化区间及函数本身的取值范围。

》fplot (fname, [xmin xmax ymin ymax], c);　　输入的第三个参数表示了函数曲线的颜色和类型。

plot 函数的使用实例：

》fplot ('sin (x) ./x', [-10*pi 10*pi],'b');

》fplot ('tan', [-pi pi -10*pi 10*pi],'b'); grid;

》fun='besselj (1, x)'; fplot (fun, [0 10*pi]); grid; title (fun); xlabel ('x');

》fun='ellipj (x, .81)'; fplot (fun, [0 3*pi]); grid; title (fun); xlabel ('x');

· fmin 函数

fmin 函数本身的功能是求一维函数极小值的位置，其用法是：

》fmin (fname, x1, x2);　　输入参数 fname 为字符串，它可以是 M 函数的名称，也可以是 MATLAB 中承认的其它数学函数或数学函数的组合，自变量的名称应该为 x；输入的另外两个参数表示了函数自变量的区间，fmin 函数就是在此区间之内找寻函数的极小值。

例 22　求函数 humps (x) 的极小值和极大值

humps 函数是专门为了演示而特意设置的函数，使用 fplot 函数可以清楚地将它的函数曲线绘制出来，如图 2.16 所示。

》fun='humps'; fplot (fun, [0 2]); grid; title ([fun ' (x)']); xlabel ('x');

从函数 bumps 的图形中可以看出来在 [0 2] 区间内，此函数有一个极小值和两个极大值。找寻函数 f (x) 的极大值可以通过找寻函数 -f (x) 的极小值的方法来得到。

》xmin=fmin ('humps', .4, .8)　　结果 xmin: 0.6370

》xmax=fmin ('-humps (x)', .2, .4)　　结果 xmax: 0.3004

》xmax=fmin ('-humps (x)', .8, 1)　　结果 xmax: 0.8927

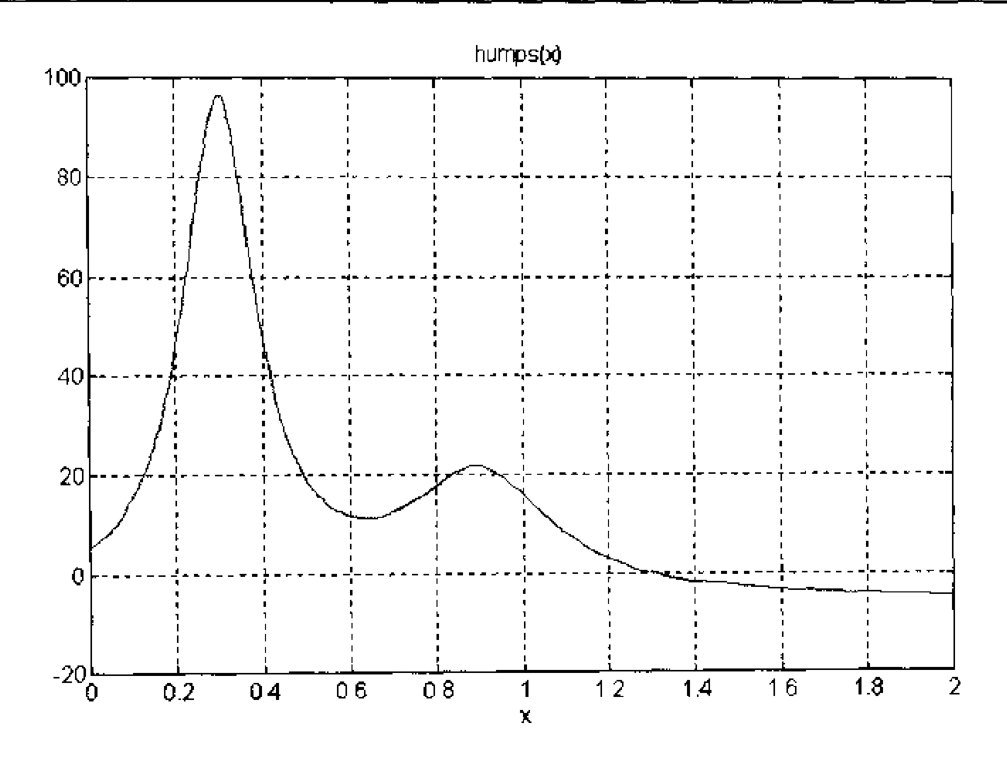


图 2.16 函数 humps 的图形

· fzero 函数

fzero 函数的功能是求一维函数零点的位置，其用法是：

〉fzero (fname, x1)；　　输入参数 fname 为字符串，它必须是 M 函数的名称，fzero 函数是在输入参数 x1 处附近找寻函数零点。

〉xz = fzero ('humps', 1)　　结果 xz：1.2995

· 数值积分

在 MATLAB 中专门设置了两个用于数值积分的函数：quad 和 quad8。quad 函数采用低阶自适应的辛普森递归方法进行数值积分，而 quad8 函数则是采用高阶自适应的牛顿－柯特斯递归方法进行数值积分。它们的用法是：

〉I = quad (fname, a, b)；

〉I = quad8 (fname, a, b)；

输入参数 fname 是字符串，它为一个 M 函数的名称，该 M 函数代表了被积函数 f (x)，a 和 b 分别为积分的上下限，即：$I = \int_a^b f(x)\,dx$。

〉I = quad8 ('humps', 0, 1.2995)；　　计算出 humps 函数在 [0 1.2995] 区间的定积分，其结果为 I：31.6981

我们知道，定积分的几何意义是以函数曲线为曲边的一个曲线梯形的面积，若要求一个椭圆的面积 (a = 2, b = 1)，先定义一个 M 函数 ellipse.m，

```
function y = ellipse (x);
global a b;
y = b * sqrt (1 - x. * x/a/a);
```

然后，键入 global a b；a = 2；b = 1；S = 4 * quad8（'ellipse'，0，a）；计算结果为 S：6.2832，这与椭圆的面积公式 S = πab 相吻合。

·常微分方程的数值解

若未知函数是一元的，那么对应的微分方程称为常微分方程，在 MATLAB 中专门设置了两个采用数值方法求解常微分方程的函数：ode23 和 ode45。ode23 函数采用二阶和三阶的龙格－库塔法来求解一阶常微分方程，而 ode45 函数则采用四阶和五阶的龙格－库塔法来求解一阶常微分方程。

采用数值法求解高阶微分方程时，必须先将其转化成为一阶微分方程组。n 阶微分方程的一般形式为：$x^n = f(t, x, x', x'', \cdots, x^{(n-1)})$

设 $x_1 = x$，$x_2 = x'$，$x_3 = x''$，…，$x_n = x^{(n-1)}$，于是得到了 n 个方程组成的一阶微分方程组：

$$
\begin{cases}
\dfrac{dx_1}{dt} = x_2 \\
\dfrac{dx_2}{dt} = x_3 \\
\cdots\cdots \\
\dfrac{dx_n}{dt} = f(t, x_1, x_2, \cdots\cdots, x_n)
\end{cases}
$$

采用数值法求解常微分方程之前，要先定义一个 M 函数将微分方程描述清楚，该 M 函数的格式如下（设函数的名称为 fdiff.m）：

```
function xp = fdiff (t, x);
xp (1) = x (2);                                % 将高阶微分方程转化
xp (2) = x (3);                                % 成一阶微分方程组
xp (n) = f (t, x (1), x (2),    x (n))         % 对微分方程进行具体描述
```

对比上面微分方程组的公式，就可以明确地了解 fdiff.m 函数的含义，在该函数内对应于 n 阶常微分方程定义了 n 个方程组，并得到了 xp 向量，它代表着 x 向量的导数，其元素数目共有 n 个。

ode23 函数和 ode45 函数的用法是：

```
》[t, x] = ode23 (fname, t1, t2, x0);
》[t, x] = ode45 (fname, t1, t2, x0);
```

输入参数 fname 为字符串，其内容是描述待解微分方程的 M 函数名，如'fdiff'。参数 t1 和 t2 给出了自变量的变化区间；而向量 x0 则表示着微分方程的初始条件，即在 $t = t_1$ 时刻，$x(t_1)$，$x'(t_1)$，…，$x^{(n-1)}(t_1)$ 的值。输出参数 t 向量代表着自变量，而输出参数 x 一般为矩阵，其第一列代表着微分方程的数值解。

采用数值法求解微分方程的关键步骤是：

1）根据所给的微分方程，建立一个特定的 M 函数，并严格地按照规定的格式来描述它。

2）根据要解决的实际问题来确定方程的初始条件，n 阶微分方程有 n 个初始条件。

下面举例说明。

例 23 一阶线性微分方程

设一阶线性微分方程为：$\frac{dy}{dx}-\frac{2y}{x+1}=(x+1)^{5/2}$，初始条件为：y (0) =1。为了采用数值法求解这个方程，先建立一个 M 函数 eq1.m，

```
function yp=eq1 (x, y);
yp= (x+1) ^2.5+2*y/ (x+1);                % 对一阶微分方程进行描述
```

然后键入 [x, y] =ode45 ('eq1', 0, 5, 1); plot (x, y); 画出的曲线与该微分方程的解析解 $y=\frac{2}{3}(x+1)^{7/2}+\frac{1}{3}(x+1)^2$ 是完全吻合的。

例 24 简谐方程 $\frac{d^2x}{dt^2}+k^2x=0$

设 $k=2\pi$，初始条件为：x (0) = 0, x′ (0) =k。先定义 M 函数 harm.m，

```
function xp=harm (t, x);
k=2*pi;
xp (1) =x (2);
xp (2) = -k*k*x (1);          % 对简谐方程进行描述
```

然后键入

```
》[t, x] =ode45 ('harm', 0, 1, [0 2*pi]);
》plot (t, x (:, 1), t, sin (2*pi*t));
```

画出的曲线与该微分方程的解析解 x (t) = sin kt 是完全吻合的。

例 25 一阶贝塞尔方程 $\frac{d^2y}{dx^2}+\frac{1}{x}\cdot\frac{dy}{dx}+(1-\frac{1}{x^2})y=0$

为了避开该方程的奇异点 x=0 处，设初始条件为：y (x1) = x1, y′ (x1) =1，其中 x1 为一个接近于 0 的正数，在 MATLAB 中不妨选择 eps 变量为 x1。定义 M 函数 bessel1.m，

```
function yp=bessel1 (x, y);
yp (1) =y (2);
yp (2) =-(1-(1/x) ^2) *y (1) -y (2) /x;    % 对一阶贝塞尔方程
进行描述
```

然后键入

```
》[x, y] =ode45 ('bessel1', eps, 10, [eps 1]);
》plot (x, y (:, 1),'b', x, besselj (1, x),'r');
```

画出的曲线与该微分方程的解析解 J_1 (x) 是线性相关的，两者间相差一个系数，其原因在于选择的初始条件与一阶贝塞尔函数及其导数的实际数值有所不同。

2.11 线性代数

MATLAB的核心是进行线性代数的运算，而这些代数与线性代数的函数放置在MATLAB的子目录elmat，matfun和polyfun内。

2.11.1 多项式函数

多项式函数包括了多项式的计算、数据插值和曲线拟合。

· roots函数

roots函数的功能是求多项式的根，此函数可以用来求解一元二次方程、一元三次方程及一元高次方程，并对多项式进行因式分解。roots函数的用法是：

》x=roots (v)；　　输入参数v为向量，它为多项式的系数，而输出参数x为列向量，它是多项式的根。

若向量v有n+1个元素，则多项式为v (1) *x^n+v (2) *x^ (n-1) +…+v (n) *x+v (n+1)。

》x=roots ([1 -2 -15])；　　计算结果x：[5 -3]′，它表示$x^2-2x-15=(x-5)(x+3)$。

》x=roots ([1 0 16])；　　计算结果x：[0+4.0i 0-4.0i]′。

》x=roots ([2 5 -76 96])；　　计算结果x：[-8 4 1.5]′，它表示$2x^3+5x^2-76x+96=(x+8)(x-4)(2x-3)$。

·poly函数

poly函数的功能是由根来确定多项式的系数，或者是求一个矩阵的特征多项式的系数，其用法是：

》v=poly (r)；　　输入参数r为向量，它表示了多项式的根，而输出参数v则是多项式的系数向量。

》v=poly (A)；　　输入参数A为矩阵，而输出参数v是A矩阵的特征多项式的系数向量，特征多项式可由行列式来表示$f(\lambda)=|\lambda I-A|$，其中I为单位矩阵，λ为参数。

· polyval函数

polyval函数的功能是用来计算多项式的值，其用法是：

》y=polyval (v, x)；　　输入参数v为多项式的系数向量，而x为该多项式中的变数，y为该多项式在x处的值。

· polyfit函数

polyfit函数的功能是多项式来进行曲线拟合，其用法是：

》v=polyfit (x, y, n)；　　向量x和y为一些数据的值，n为进行曲线拟合的多项式P (x) 的项数，v为该多项式的系数向量，这

些系数 在满足最小二乘的条件下，使 P（x_i）≅y_i。

例 26 使用多项式进行曲线拟合

```
》x=0：7；  y=sin（x）；  xx=0：.1：7；  % 生成一些数据
》v7=polyfit（x，y，7），v5=polyfit（x，y，5），v3=polyfit（x，y，3），  %多项式
》y7=polyval（v7，xx）；y5=polyval（v5，xx）；y3=polyval（v3，xx）；
》whitebg（'w'）；plot（x，y，'ko'，xx，y7，'b'，xx，y5，'r'，xx，y3，'m'）；
```

从曲线拟合的结果看来，5 次和 7 次多项式的效果相差不多，而 3 次多项式的效果较差。三个多项式的系数向量为

v7：[0.0001 −0.0031 0.0211 −0.0286 −0.1328 −0.0188 1.0035 0]

v5：[−0.0048 0.0756 −0.3347 0.1448 0.9697 −0.0016]

v3：[0.0656 −0.6388 1.3649 0.0428]

· polyder 函数

polyder 函数的功能是计算多项式的导数，其用法是：

》vp=polyder（v）； 向量 v 为多项式的系数向量，而向量 vp 为该多项式的导数。

》vp=polyder（A，B） 与》v=conv（A（:），B（:））；vp=polyder（v）等价。

· residue 函数

residue 函数的功能是进行部分分式展开（残数或留数计算），设 B（s）和 A（s）为多项式，在不存在重根的情况下，B（s）/A（s）的部分分式为

$$\frac{B(s)}{A(s)}=\frac{r_1}{s-p_1}+\frac{r_2}{s-p_2}+\cdots+\frac{r_n}{s-p_n}+K(s)$$

其中，r_i称为残数（留数），p_i 称为极点，多项式 K（s）称为直接项。

residue 函数的用法是：

》[r，p，K]=residue（B，A）； B 为分子多项式的系数向量，A 为分母多项式的系数向量；r 为残数向量，p 为极点向量，K 为直接项的系数向量。

· interp1 函数

interp1 函数的功能是进行一维数据的插值，其用法是：

》yi=interp1（x，y，xi，ms）； x 和 y 向量代表了原始数据，而 xi 和 yi 向量则代表了一维数据插值以后的结果，其中 xi 向量由 x 向量等间距插值而成，ms 为字符串，它代表着插值的方法。

ms='linear' 为线性插值，ms='spline'为三次样条插值，ms='cubic'为三次插 值。线性插值为缺省的方法。

·spline 函数

spline 函数的功能是进行一维数据的三次样条插值。样条函数是函数逼近的一种方法，

三次样条插值采用的是三次多项式，其主要特点是具有连续的一阶导数和二阶导数，因此插值以后的曲线是十分光滑的。spline 函数的用法是：

》yi=spline（x，y，xi）；　　x 和 y 向量代表了原始数据，而 xi 和 yi 向量则代表了三次样条插值以后的结果。

三次样条插值的一个实例：

》x=0：10；y=sin（x）；xi=0：.25：10；yi=spline（x，y，xi）；

》whitehg（'w'）；plot（x，y，'ko'，xi，yi，'b'）；

2.11.2 基本矩阵

· rand 函数

rand 函数的功能是产生在（0，1）区间内均匀分布的随机数矩阵，由于每次进入 MATLAB 工作环境时，产生随机数使用的种子是相同的，也就是说随机数的产生是可以重复的，例如在某个函数中使用了 rand 函数，那么在每次进入 MATLAB 后调用该函数，就会得到相同的随机数序列。rand 函数多用来模拟噪声，因此在进行电子仿真的过程中，它的作用是十分重要的，其用法是：

》x=rand（3）；　　生成 3×3 均匀分布的随机数矩阵。

》x=rand（m，n）；　　生成 m×n 均匀分布的随机数矩阵。

》x=rand（size（A））；　　生成与 A 矩阵大小相同的均匀分布的随机数矩阵。

》s=rand（'seed'）　　得到当前产生随机数的种子，s：931316785 。

》rand（'seed'，s）　　将 s 的数值作为种子。

》rand（'seed'，0）　　恢复种子的缺省数值。

》rand（'seed'，sum（100 * clock））　　将种子与时间联系起来以保证每次进入 MATLAB 时产生的随机数不同。

·randn 函数

randn 函数的功能是产生正态分布的随机数矩阵，其均值为 0，方差为 1，它的用法与 rand 函数相同。

·linspace 函数

linspace 函数的功能是产生均匀间隔的向量，其用法是：

》x=linspace（x1，x2，n）；　　在 x1 与 x2 间均匀生成 n 个元素组成的向量。

·logspace 函数

logspace 函数的功能是产生对数间隔的向量，其用法是：

》x=logspace（d1，d2，n）；　　在 10^d1 与 10^d2 间生成 n 个元素组成的向量，其间隔为对数关系。

· clock 函数

clock 函数的功能是产生 1×6 的时间向量，输出格式为 [年 月 日 时 分 秒]，其用法是：

》t = fix (clock)　　　　输出结果为 [2000 2 11 18 55 45]。

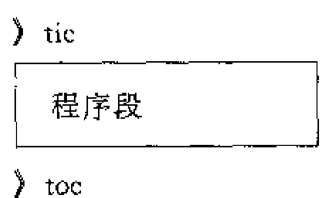
·date 函数

date 函数的功能是产生一个表示日期的字符串，其输出格式为 dd - mmm - yy。

·秒表

tic 函数用于启动一个模拟的秒表，而 toc 函数则用来读秒表，两个函数联合就可以计算某个程序运行的时间，

》tic



程序段

》toc　　　　%显示所用时间

· diag 函数

diag 函数的主要功能是生成一个对角矩阵，或将一个矩阵的对角线上的元素提取出来，其用法是：

》A = diag (v)	v 为向量，A 为对角矩阵，其对角线上的元素由 v 向量表示。
》A = diag (v, k);	k>0 时，A 矩阵对角线右上方的第 k 条次对角线元素由 v 向量表示；k<0 时，A 矩阵对角线左下方的第 k 条次对角线元素由 v 向量表示。
》v = diag (A)	v 为向量，它由 A 矩阵对角线上的元素组成。
》v = diag (A, k)	v 为向量，它由 A 矩阵中第 k 条次对角线上的元素组成。
》B = diag (diag (A))	B 为对角矩阵，它由 A 矩阵对角线上的元素组成。

如：A = diag (1: 3);　B = diag (1: 3, 1);　C = diag (1: 3, -2);　结果为

$$A=\begin{matrix}1&0&0\\0&2&0\\0&0&3\end{matrix}\quad B=\begin{matrix}0&1&0&0\\0&0&2&0\\0&0&0&3\\0&0&0&0\end{matrix}\quad C=\begin{matrix}0&0&0&0&0\\0&0&0&0&0\\1&0&0&0&0\\0&2&0&0&0\\0&0&3&0&0\end{matrix}$$

·fliplr 函数

fliplr (A) 的功能是对 A 矩阵进行左右翻转，即第一列成为最后一列，最后一列成为第一列。

· flipup 函数

flipup (A) 的功能是对 A 矩阵进行上下翻转，即第一行成为最后一行，最后一行成为第一行。

·rot90 函数

rot90（A）的功能是将 A 矩阵沿逆时针方向旋转 90°，即第一行成为第一列，第一列成为最后一行。

· reshape 函数

reshape 函数的功能是对一个矩阵的元素按列进行重组，以便对矩阵的大小进行改变，其用法是：

》B＝reshape（A，m，n）　　A 矩阵的元素数目必须是 m×n 个。如 A 矩阵为 4×4 矩阵，则 B 矩阵可以是 2×8 矩阵，也可以是 8×2 矩阵。

· tril 函数

tril 函数的功能是提取下三角矩阵，其用法是：

》B＝tril（A）；　　B 矩阵的元素取自 A 矩阵对角线以下的元素（含对角线）。

》B＝tril（A，k）；　　k>0 时，B 矩阵的元素取自 A 矩阵对角线上方第 k 条次对角线以下的元素（含次对角线），k<0 时，B 矩阵的元素取自 A 矩阵对角线下方第 k 条次对角线以下的元素（含次对角线）。

· triu 函数

triu 函数的功能是提取上三角矩阵，其用法与 tril 函数相仿。

2.11.3 特殊矩阵

在 MATLAB 中提供了若干特殊矩阵，见表 2.8，若想进一步了解这些矩阵的使用，可以键入 help specfun 来启动联机帮助文件。

表 2.8　特殊矩阵一览表

特殊矩阵	说　明
compan（p）	多项式（系数向量为 p）的伴随矩阵
gallery	几个小的测试矩阵
hadamard（n）	n 阶的阿达玛矩阵
hankel（c）	汉克尔矩阵，c 向量为其第一列元素
hilb（n）	n×n 的希尔伯特矩阵
invhilb（n）	n×n 的希尔伯特矩阵的逆矩阵
kron（A，B）	克罗内克尔张量积
magic（n）	n×n 的魔方矩阵
pascal（n）	n×n 的帕斯卡矩阵
rosser	一个经典的对称特征根的测试矩阵
toeplitz（c，r）	托埃普利茨矩阵
vander（c）	范德蒙矩阵
wilkinson（n）	威尔金森的特征值测试矩阵

2.11.4 矩阵函数

在 MATLAB 中提供了若干矩阵函数，使用数值方法解决线性代数的问题，有兴趣的读者可键入 help matfun 来启动联机帮助文件，具体了解这些函数的用法。此处仅对一些常用的函数进行简单的介绍。

· rank 函数

rank（A）的功能是求 A 矩阵的秩。

· det 函数

det（A）的功能是计算 A 矩阵的行列式的值，A 矩阵应该是方阵。

· eig 函数

eig 函数的功能是计算矩阵的特征根和特征向量，其用法是

》v=eig（A）;　　　v 为列向量，其元素为 A 矩阵的特征根。

》[B，C] =eig（A）;　　　B 矩阵的列向量为 A 矩阵的特征向量，而 C 为对角矩阵，其对角线上的元素为 A 矩阵的特征根

参考文献

[1]（美）Duane Hanselman，Bruce Littlefield. 精通 MATLAB. 西安：西安交通大学出版社，1988

[2] 张培强．MATLAB 语言．合肥：中国科技大学出版社，1995

[3] 薛定宇．控制系统计算机辅助设计．北京：清华大学出版社，1996

[4]（美）帕普里斯 A．信号分析．北京：科学出版社，1981

第三章 MATLAB的高级操作

在本章内介绍 MATLAB 的一些高级的操作，其中包括句柄的使用、绘制三维图形、声音和图象的处理、菜单式界面和按钮式界面的编程，读者掌握了这些方法之后可以较为得心应手地使用 MATLAB 编制一些实用的程序。

3.1 句柄与对象属性

在 MATLAB 中，句柄（handle）给使用者提供了对图形和对象进行底层控制的功能，借助于句柄人们可以根据需要对图形和对象的参数进行调整，而在高级命令和 M 函数中一般是不提供上述功能的。

由于句柄本身是比较复杂的，因此它属于 MATLAB 高级操作的范畴。

3.1.1 一般概念

句柄大多是与图形联系在一起的。一幅图形实际上是由若干个图形元素构成的，这些图形元素包括曲线、文字、坐标轴等。每一个图形元素就是一个对象（object），每一个对象都有一个句柄与它相关，因此通过句柄可对某个对象进行修改，而不会影响其它的对象。

对象是计算机软件范畴内的一个概念，对象是由一组紧密相关、形成唯一整体的数据结果或函数的集合。在 MATLAB 中，按照隶属关系对象可以分为四层：

1）第一层是根对象，根对象就是计算机的屏幕，根对象只有一个。

2）第二层是图形窗口对象，图形窗口对象可以有若干个，它们之间是并列的关系，当屏幕上出现多个图形窗口时，就存在着多个图形窗口对象。

3）第三层对象包括坐标对象、用户菜单对象和用户控制框对象，它们之间的关系也是并列的，同时也可以有多个对象同时存在。

4）第四层对象包括文本对象、线条对象、图象对象、贴片对象和曲面对象，它们之间的关系也是并列的，同时也可以有多个对象同时存在。

上一层的对象称为父对象，面下一层的对象称为子对象。由于对象的内容是十分复杂的，故在本节内仅介绍经常要进行处理的图形对象、坐标对象、线条对象和文本对象。

每个句柄都有一个数字与它对应，这个数字起着标识的作用，此数字可以称为是句柄的数值，可以赋值给某个变量，为了与其它的变量区别，句柄的变量名通常以大写字母 H 打头。根对象句柄的数值为 0，图形窗口句柄的数值为整数，其它句柄的数值则都是浮点数。图形窗口的句柄是通过 gcf 函数得到的，坐标系的句柄是通过 gca 函数得到的，gcf 函数和 gca 函数输出的就是句柄的数值；而其它对象的句柄则是通过相应函数的输出参数而得到的。例如：

》H = plot（x，y）；　　　　H 就是 plot 函数的句柄。

》H1 = text (1, 2,'bbi'); H1就是text函数的句柄。

一个句柄代表着一个对象，而一个对象有着若干属性，对象属性由属性名和属性值两部分组成。属性名为一个特定的字符串，而属性值则有数量、向量、矩阵、字符串和字符串矩阵等几种类型。改变对象属性可以对一个对象的相应部分进行修改，而不会影响其它的对象。在MATLAB中，属性名是不分大小写的，而每个对象属性都有其特定的缺省值。

在MATLAB中，获得和改变对象的属性是通过get函数和set函数来完成的。

· get函数

get函数的功能是获取对象的属性，其用法是：

》get (H); 输入参数H为对象的句柄，输出的是该对象所有的属性名和属性值。

》get (H, p); 字符串p为一个属性名，输出的是该属性的值。

》get (0); 获取根对象的所有属性。

》get (gcf); 获取当前图形窗口的所有属性。

》get (gca); 获取当前坐标系的所有属性。

·set函数

set函数的功能是改变对象的属性，其用法是：

》set (H, p, v); H为句柄，字符串p为一个属性名，v是给该属性新设定的值。

》set (H, p1, v1, p2, v2); 改变两个属性的值。

》set (H, p); 显示一个属性所有的可能值。

》set (H); 显示一个对象所有的属性，及其所有的可能值。

·reset函数

reset (H) 的作用是恢复对象属性所有的缺省值。

·delete函数

delete (H) 的作用是删除一个对象及其子对象，但不影响其父对象和与其并列的对象。

· gco函数

gco函数的作用是获取当前对象的句柄，而当前对象是指刚刚用鼠标点击过的对象，若未使用过鼠标点击某个对象，则返回的是空矩阵。

3.1.2 图形窗口对象

在MATLAB中，所有的绘图函数都会自动打开一个图形窗口，还有一些函数如：text、xlabel、title、axis等也可以自动打开一个图形窗口，另外还有一个专门用来建立图形窗口的函数，称为figure函数。

》get (gcf); 便获取了当前图形窗口的所有属性。不同版本的MATLAB的图形窗口对象的属性是不同的，4.2版本的典型结果为：

```
BackingStore = on
Color = [0 0 0]
Colormap = [ (64 by 3) ]
CurrentAxes = [79.0005]
CurrentCbaracter = g
CurrentMenu = [1]
CurrentObject = []
CurrentPoint = [0 0]
FixedColors = [ (8 by 3) ]
InvertHardcopy = on
KeyPressFcn =
MenuBar = figure
MinColormap = [64]
Name =
NextPlot = add
NumberTitle = on
PaperUnits - inches
PaperOrientation = portrait
PaperPosition = [0.25 2.5 8 6]
PaperSize = [8.5 11]
PaperType = usletter
Pointer = arrow
Position = [120 120 560 420]
Resize = on
SelectionType = normal
ShareColors = yes
Units = pixels
WindowButtonDownFcn =
WindowButtonMotionFcn =
WindowButtonUpFcn =

ButtonDownFcn =
Children = [79.0005]
Clipping = on
Interruptible = no
Parent = [0]
Type = figure
UserData = []
Visible = on
```

从以上的结果可以看出，图形窗口的对象属性是相当多的，而对于本书的读者来说，并不需要详细的了解每一个对象属性，因此在此处仅对几个常用的图形窗口对象属性进行简单的介绍。

·Color（背景颜色）

背景颜色是指图形窗口的背景而言的，实际上 whitebg 函数就是通过此对象属性对图形的背景颜色进行控制的。背景颜色的属性值是 1×3 的向量 [r g b]，即用红绿蓝三个数值表示的。

》set (gcf,'Color', [.2 .4 .4]);　　将图形窗口的背景颜色设置为暗的蓝绿色。

对于 5.x 版本的 MATLAB 来说，背景颜色是指坐标轴以外的区域而言的，缺省值为 [.8 .8 .8] 为灰色背景。

·Name（图形窗口名称）

每个图形窗口都可以有一个名称，它是位于图形窗口编号后面的。若使用者想给某个窗口起一个名字，则可以通过 Name 属性的属性值进行赋值来完成。

》set (gcf,'Name', s);　　s 为一个字符串，它就成为该窗口的名称。

·NumberTitle（图形窗口编号）

每个图形窗口都有一个编号，在图形窗口的上方有 Figure No.x 的字样，其中 x 就是窗口的编号，编号是从 1 开始累加的。

NumberTitle 属性的属性值有两个。通过 set（gcf,'NumberTitle'）；就得到了属性值为 [{on} | off]。缺省的属性值为 on，它是用大括号内的字符串来表示的。

属性名可以简化为 Num。

》set（gcf,'NumberTitle','off'）；　　就去除了图形窗口上的 Figure No.x 字样

· Units（长度单位）

在确定图形窗口的尺寸时就一定会涉及到长度的单位问题。Units 的属性值共有 5 个，它们是 [inches | centimeters | normalized | points | {pixels}]，其含义依次为英寸、厘米、归一值、点（磅）和象素，缺省值是象素。

通常计算机屏幕的设置为 640×480、800×600、1024×768，这些就是以象素为单位的，因此使用象素是比较方便的。点（磅）为印刷使用的单位，1 点等于 1/72 英寸，约为 0.353 毫米。归一值是个相对的概念，其最大值为 1，也就是说屏幕宽度和高度均为 1。

属性名可以简化为 Uni。

·Position（位置）

位置属性表明了图形窗口的具体位置和大小（包括横向和纵向两个方向），属性值为一个向量：[left bottom width height]，其含义依次是左、下方、宽、高。左和下方两个参数表明了图形窗口的左下角在计算机屏幕上的具体位置，宽和高的含义则是一目了然的，它们表明了图形窗口的尺寸。位置属性所采用的单位由 Units 属性来确定。当计算机屏幕的设置为 800×600 时，位置属性的缺省值是 [120　120　560　420]。

属性名可以简化为 Pos

·Resize（调整窗口大小）

Resize 属性表明了是否允许改变图形窗口的尺寸，其属性值有 2 个 [{on} | off]。缺省值为 on，表示允许使用鼠标来改变图形窗口的尺寸。

属性名可以简化为 Res。

·figure 函数

figure 函数的作用是打开一个新的图形窗口，其用法是：

》H=figure;　　开一个新的图形窗口，并返回一个整数句柄 H。

》H=figure（p1，v1，p2，v2，p3，v3，…）；　　打开一个新的图形窗口，返回一个整数句柄 H，同时对图形窗口对象的若干属性进行设定（不采用缺省值）。

在第二章第六节中介绍的 specanl 函数中，读者可以清楚地看出使用 figure 函数所起的作用，它分别打开了模拟示波器和模拟频谱仪两个窗口，并对窗口的位置、窗口名称属性等进行了设定。

· close 函数

close 函数的作用是关闭图形窗口，其用法是：

》close;　　　　关闭当前的窗口。

》close (H);　　　关闭句柄 H 表示的窗口。

》close ('name');　关闭名为 name 的窗口。

》close all;　　　关闭所有的窗口。

3.1.3 坐标对象

在 MATLAB 中，所有的绘图函数都会自动建立一个坐标，另外还有一个专门用来建立坐标的函数，称为 axes 函数。坐标对象是图形窗口对象的子对象。

》get (gca); 便获取了当前图形窗口中坐标的所有属性。不同版本的 MATLAB 的坐标对象的属性是不同的，4.2 版本的典型结果为：

```
AspectRatio = [NaN NaN]
Box = on
CLim = [0 1]
CLimMode = auto
Color = none
CurrentPoint = [ (2 by 3) ]
ColorOrder = [ (6 by 3) ]
DrawMode = normal
FontAngle = normal
FontName = Helvetica
FontSize = [12]
FontStrikeThrough = off
FontUnderline = off
FontWeight = normal
GridLineStyle = :
LineStyleOrder = -
LineWidth = [0.5]
NextPlot = replace
Position = [0.13 0.11 0.775 0.815]
TickLength = [0.01 0.025]
TickDir = in
Title = [63.0001]
Units = normalized
View = [0 90]
XColor = [1 1 1]
XDir = normal
Xform = [ (4 by 4) ]
XGrid = off
XLabel = [60.0002]
XLim = [0 120]
XLimMode = auto
XScale = linear
XTick = [ (1 by 7) ]
XTickLabels =
  0
  20
  40
  60
  80
  100
  120
XTickLabelMode = auto
XTickMode = auto
YColor = [1 1 1]
YDir = normal
YGrid = off
YLabel = [61.0002]
YLim = [0 120]
YLimMode = auto
YScale = linear
YTick = [ (1 by 7) ]
YTickLabels =
```

```
 0
 20
 40
 60
 80
 100
 120
YTickLabelMode = auto
YTickMode = auto
ZColor = [1 1 1]
ZDir = normal
ZGrid = off
ZLabel = [62.0002]
ZLim = [-1 1]
ZLimMode = auto
ZScale = linear
ZTick = [-1 0 1]
ZTickLabels =
ZTickLabelMode = auto
ZTickMode = auto

ButtonDownFcn =
Children = [59.0002]
Clipping = on
Interruptible = no
Parent = [1]
Type = axes
UserData = []
Visible = on
```

与讨论窗口图形对象的情况类似，在此处仅对几个常用的坐标对象属性进行简单的介绍。

·AspectRatio（图象尺寸比）

在绘图过程中，图象尺寸比是一个重要的坐标属性，特别是在绘制三维图形的时候，往往要对此项属性进行设置。图象尺寸比的属性值是个向量 [axis-ratio data-ratio]，前面的一个元素为图象的宽高比（w/h），后面的元素为水平方向和垂直方向的数据单位之比，缺省值为 [NaN NaN]。

属性名可简化为 Asp。

》axis equal　对于二维图形来说，实际上就相当于　》set (gca,'asp', [NaN 1]);

》axis image　实际上就相当于　》set (gca,'asp', [1　1]);

· Color（颜色）

坐标对象的颜色属性是指坐标轴以内的区域而言的，属性值为向量 [r g b]，即用红绿蓝三个数值表示的。4.2 版本的缺省值是 none，意为没有颜色；5.x 版本的缺省值是 [1 1 1]，为白色。

· FontAngle（字体角度）

字体角度包括了坐标轴、x 轴说明、y 轴说明、z 轴说明、标题上的字体角度。字体角度的属性值有 3 个 [{normal} | italic | oblique]，其含义依次为正常、斜体、斜体（某些系统中），缺省值为正常。

属性名可以简化为 FontA。

·FontName（字体）

字体是指坐标轴、x 轴说明、y 轴说明、z 轴说明、标题上的字体，缺省的字体为 Helvetica，通常还可选用 arial，courier，times，symbol 等字体。

属性名可以简化为 FontN。

· FontSize（字体大小）

字体大小包括了坐标轴、x 轴说明、y 轴说明、z 轴说明、标题上的字体大小。字体大小的单位是点（0.353mm），缺省值是 12 点（相当于四号字）。

属性名可以简化为 FontS。

·FontUnderline（字体下划线）

字体下划线的属性值有 2 个 [on | {off}]，缺省值为 off，意为没有下划线。属性名可以简化为 FontU。

· FontWeight（字体深浅）

字体深浅是指字体颜色的深浅，其属性值有 4 个 [light | {normal} | demi | bold]，其含义依次为浅色字、正常、适中、黑体字，缺省值为正常。

属性名可以简化为 FontW。

· GridLineStyle（栅格类型）

栅格类型属性值有 4 个 [- | -- | {:} | -.]，其含义依次为实线、虚线、点线、点划线，缺省值为点线。

属性名可以简化为 Grid。

·LineWidth（坐标轴宽度）

坐标轴宽度是指构成坐标轴的直线的宽度，其单位为点，缺省值为 0.5 点。属性名可以简化为 LineW。

· Units（长度单位）

坐标对象的长度单位与图形窗口对象的长度单位含义是一样的，所不同是坐标对象长度单位的缺省值是归一值。

· Position（位置）

位置属性表明了坐标轴在图形窗口的具体位置和大小，注意此处的坐标轴是指水平坐标和垂直坐标而言的。坐标位置的属性值为一个向量：[left bottom width height]，其含义依次是左、下方、宽、高。左和下方两个参数表明了坐标的左下角在图形窗口中的具体位置，[0 0] 为图形窗口的左下角，[1 1] 为图形窗口的右上角；宽表明了水平坐标的长度，而高则代表着垂直坐标的长度。位置属性所采用的单位由 Units 属性来确定。

属性名可以简化为 Pos。

· Title（标题）

标题属性的属性值就是由 title 函数生成的标题的句柄，通过这个句柄可以改变标题的格式。标题对象是文本对象的一种。

》H-title = get（gca, 'title'）； get（H-title）； 获得了标题对象的属性。

· View（视角）

视角属性是指观察者的视角，在三维绘图过程中，视角的作用是十分重要的，因为计算机屏幕是二维的，要在上面显示三维图形就必须根据一定的视角将三维图形变换成二维图形。当视角改变以后，三维图形的外观也随之改变。

视角属性的属性值为向量［az el］，其中 az 称为方位角，在 MATLAB 中方位角是 在 xy 平面上从 −y 轴开始向右计算的，即 az = 0°对应着 −y 轴，而 az = 90°对应着 x 轴；el 称为仰角，在 MATLAB 中仰角是从 xy 平面向上计算的，即 el = 0°对应着 xy 平面，而 el = 90°对应着 +z 轴。关于方位角和仰角的其它情况，见本章的第四节。

在二维图形中，视角的缺省值是［0 90］，而在三维图形中，视角的缺省值是［−37.5 30.0］。

·XColor（x 轴颜色）

x 轴颜色包括了 x 轴本身的颜色、与 x 轴平行的栅格的颜色及 x 轴刻度字符的颜色，属性值为向量［r g b］。YColor 和 ZColor 与 Xcolor 类似。

·XDir（x 轴方向）

x 轴方向表明了 x 值增加的方向，其属性值有 2 个［normal | {reverse}］，即正向、反向，缺省值是正向。YDir 和 ZDir 与 XDir 类似。

· XGrid（x 轴栅格）

x 轴栅格表明了沿 x 方向是否有栅格，其属性值有 2 个［on | {off}］。缺省值是 off，表示没有栅格。grid 函数就是对这个属性进行控制。YGrid 和 ZGrid 与 XGrid 类似。

·XLabel（x 轴说明）

x 轴说明的属性值就是由 xlabel 函数生成的句柄，通过这个句柄可以改变说明的格式。x 轴说明对象是文本对象的一种。

》Hx = get（gca, 'XLalel'）； get（Hx）； 获得了 x 轴说明对象的属性。

YLabel 和 ZLabel 与 XLabel 类似。

·XLim（x 轴区间）

x 轴区间表明了 x 的取值范围，其属性值为向量［xmin xmax］。axis 函数就对这个属性进行控制。YLim 和 ZLim 与 XLim 类似。

·XScale（x 轴尺度）

x 轴尺度的属性值有 2 个 [{linear} | log]，其含义是线性、对数，缺省值是线性。loglog 函数、semilogx 函数和 semilogy 函数就对这个属性进行控制。YScale 和 ZScale 与 XScale 类似。

· XTick（x 轴刻度）

x 轴刻度的属性值为一个向量，它代表了 x 轴上刻度的数值；若此属性值为一个空矩阵，那么在 x 轴上就不存在刻度。YTick 和 ZTick 与 XTick 类似。

· XTickLabel（x 轴刻度字符）

x 轴刻度字符的属性值为一个字符矩阵，它把 x 轴刻度的数值变换称为字符串，坐标上显示的数字就是此字符矩阵的内容。

YTickLabel 和 ZTickLabel 与 XTickLabel 类似。

· axes 函数

axes 函数的功能是建立一个坐标对象，其用法是：

》H = axes（'position'，[left bottom width height]）；根据所给的位置向量，在指定位置建立一个坐标系，并返回一个句柄 H。位置向量的含义与坐标位置属性中的一样，缺省的单位为归一值。

》H = axes（p1，v1，p2，v2，p3，v3，…）；　建立一个坐标系，并返回一个句柄 H，同时对坐标对象的若干属性进行设定（不采用缺省值）。

3.1.4 线条对象

在 MATLAB 中，绘图函数建立的对象大多包含了线条对象，另外还有一个专门用来建立线条的函数，称为 line 函数。线条对象是坐标对象的子对象，是最底层的对象。

》H = plot（x，y）；　get（H）　便获取了当前线条对象的所有属性。不同版本的 MATLAB 的线条对象的属性是不同的，4.2 版本的典型结果为：

```
Color = [1 1 0]                              Children = []
EraseMode = normal                           Clipping = on
LineStyle = -                                Interruptible = no
LineWidth = [0.5] MarkerSize = [6]           Parent = [84.0006]
Xdata = [ (1 by 111) ]                       Type = line
Ydata = [ (1 by 111) ]                       UserData = []
Zdata = []                                   Visible = on
ButtonDownFcn =
```

· Color（颜色）

线条对象的颜色是指线条本身的颜色而言的，属性值为向量 [r g b]，即用红绿蓝三个数值表示的。通过颜色的属性值，可以任意地改变线条的颜色，而不受绘图函数中颜色参数的限制。

· LineStyle（线条类型）

线条类型属性值有 9 个 [{-|| --|:|-.|+|o|*|.|x]，前 4 个为线条的类型，依次为实线、虚线、点线、点划线，后 5 个为标记，缺省值为实线。

属性名可以简化为 LineS。

· LineWidth（线条宽度）

线条宽度的单位为点（0.353mm），缺省值为 0.5 点。属性名可以简化为 LineW。

· MarkerSize（标记大小）

标记的单位为点（0.353mm），缺省值为 6 点。属性名可以简化为 Mark。

· Xdata（X 数据）

X 数据就是线条在 x 坐标上的数据向量。属性名可以简化为 X。

· Ydata（Y 数据）

Y 数据就是线条在 y 坐标上的数据向量。属性名可以简化为 Y。

· Zdata（Z 数据）

Z 数据就是线条在 z 坐标上的数据向量。属性名可以简化为 Z。

· line 函数

line 函数的功能是建立一个线条对象，其用法是：

》H=line（x，y，p1，v1，p2，v2，p3，v3，…）；　建立一个线条对象，并返回一个句柄 H，同时对坐标对象的若干属性进行设定（不采用缺省值）。

》H=line（x，y，z，p1，v1，p2，v2，p3，v3，…）；　建立一个三维的线条对象，并返回一个句柄 H，同时对坐标对象的若干属性进行设定。

3.1.5 文本对象

在 MATLAB 中，text 函数建立的对象为文本对象，另外 xlabel 函数、ylabel 函数、zlabel 函数、title 函数建立的对象也是文本对象。文本对象是坐标对象的子对象，是最底层的对象。

》H = text (x, y);　get (H);　便获取了当前文本对象的所有属性。不同版本的MATLAB的线条对象的属性是不同的，4.2版本的典型结果为：

```
Color = [1 1 1]                                   Rotation = [0]
EraseMode = normal                                String = bbi
Extent = [0.198614 0.469208 0.0508083             Units = data
          0.0527859]                              VerticalAlignment = middle
FontAngle = normal                                ButtonDownFcn =
FontName = Helvetica                              Cbildren = []
FontSize = [12]                                   Clipping = off
FontStrikeThrough = off                           Interruptible = no
FontUnderline = off                               Parent = [96.0006]
FontWeight = normal                               Type = text
HorizontalAlignment = left                        UserData = []
Position = [0.2 0.5 0]                            Visible = on
```

在文本对象中，FontAngle、FontName、FontSize、FontUnderline、FontWeight 等属性的含义与坐标对象中相应属性的含义是类似的，故不在此处重复。

·Color（颜色）

文本对象的颜色是指文本本身的颜色而言的，属性值为向量 [r g b]，即用红绿蓝三个数值表示的。通过颜色的属性值，可以任意地设定文本的颜色。

·HorizontalAlignment（水平方向对齐）

水平方向对齐属性规定了沿水平方向上文本对齐的方式，其属性值有 3 个 [{left} | center | right]，其含义依次为左对齐、中间对齐、右对齐，缺省值是左对齐。

属性名可以简化为 H。

· Position（位置）

位置属性就是 text 函数输入的文本位置参数，其属性值为向量 [x y z]，单位由 Units 属性指定。

属性名可以简化为 Pos。

·Rotation（旋转）

旋转属性的属性值是指文本沿逆时针方向旋转的角度，单位为度，其缺省值为 0，表示文本为水平方向。

属性名可以简化为 Rot。

· String（字符串）

字符串就是要显示的文本，属性名可简化为 S。

· Units（位置单位）

位置单位的属性值共有 6 个［inches | centimeters | normalized | points | pixels | {data}］，其含义依次为英寸、厘米、归一值、点（磅）、象素、数据（即当前坐标上的位置），缺省值是数据。

· VerticalAlignment（垂直方向对齐）

垂直方向对齐属性规定了沿垂直方向上文本对齐的方式，其属性值有 5 个［top | cap | {middle} | baseline | bottom］，其含义依次为顶部对齐、依大写字母顶部对齐、中部对齐、字母基线对齐、底部对齐，缺省值是中部对齐。

属性名可以简化为 V。

3.2 声音处理

在 MATLAB 中设置了几个声音处理函数，以支持对声音进行处理，这些函数的功能可以完成对声音文件的读写，并可把声音数据转换成为声音。需要说明的是，在声音处理方面，4.2 版和 5.x 版的 MATLAB 之间存在较明显的差别。

· sound 函数

sound 函数的功能是将表示声音的数据（声音向量）转换成为音频信号。即完成发声的工作。在 4.2 版本中，该函数只支持单声道，其用法如下：

》sound（y，fs）；　　y 为声音向量，fs 是 y 向量的采样频率，单位为 Hz。sound 函数对 y 向量的数值进行自动控制，以适应计算机硬件对输入信号的要求。

》sound（y）；　　y 为声音向量，采样频率为缺省值 8192Hz。

在 5.x 版本中，sound 函数支持双声道，其用法如下：

》sound（y，fs）；　　y 为声音向量或矩阵，其数值范围是［-1 1］，超出该范围的信号自动被削波。当 y 为 N×2 矩阵时，声音为立体声，y 的第一列为右声道，y 的第二列为左声道。fs 是采样频率，单位为 Hz。

》sound（y）；　　y 为声音向量或矩阵，采样频率为缺省值 8192Hz。

》sound（y，fs，bits）；　　y 为声音向量或矩阵，fs 是采样频率，单位为 Hz。bits 为数据量化的比特数，2≤hits≤16。

在 5.x 版本中，还有一个 soundsc 函数，其用法与 sound 函数基本相同，只是 y 矩阵的数值范围不受到限制，这点与 4.2 版本中的 sound 函数类似。

· wavwrite 函数

wavwrite 函数的功能是将声音数据按指定格式存入微软的 wav 文件中。在 4.2 版本中，该函数仅支持单声道信号，其用法如下：

》wavwrite（k，fs，file）；　　k 为 8 比特量化以后的声音向量，其元素为［0 255］区

间范围内的整数，fs 是采样频率，单位为 Hz，file 为 wav 文件的文件名，它是个字符串，后缀 .wav 自动加在文件名的后面。

在 5.x 版本中，wavwrite 函数支持双声道信号，其用法如下：

》wavwrite (k, fs, nbits, file);　　y 为声音向量或矩阵，它的数值范围是 [- 1 1]，超出该范围的信号自动被削波。fs 是采样频率，单位为 Hz。nbits 为数据量化的比特数，nbits × 16。file 为 wav 文件的文件名，它是个字符串，后缀 .wav 自动加在文件名的后面。

》wavwrite (k, fs, file);　　nbits = 16 (缺省值)。

》wavwrite (k, file);　　fs = 8000Hz (缺省值)，nbits = 16 (缺省值)。

· wavread 函数

wavread 函数的功能是从微软的 wav 文件中读取声音信号。在 4.2 版本中，该函数仅支持单声道信号，其用法如下：

》[y, fs] = wavread (file);　　file 为字符串，它表示 wav 文件的文件名，后缀可以省略；y 为声音向量，fs 为采样频率，单位为 Hz。此函数只支持 8 比特的单声道信号。

在 5.x 版本中，wavwrite 函数支持双声道信号，其用法如下：

》[y, fs, nbits] = wavread (file); file 为字符串，它表示 wav 文件的文件名，后缀可以省略；y 为声音向量，fs 为采样频率，单位为 Hz，nbits 为数据量化的比特数。

· mu2lin 函数

mu2lin 函数将按照 μ 律压缩编码的信号转换成为线性编码信号，其用法是：

》y = mu2lin (mu);　　mu 是按照 μ 律压缩编码的信号，其数值是在 [0 255] 范围内的整 数；y 是输出的线性编码信号，其数值范围是 [- .9803 .9803]。

· lin2mu 函数

lin2mu 函数将线性编码信号转换成为按照 μ 律压缩编码的信号，其用法是：

》mu = mu2lin (y);　　y 是输入的线性编码信号，其数值范围是 [- 1 1]；mu 是按照 μ 律压缩编码的信号，其数值是在 [0 255] 范围内的整数；

·auwrite 函数

auwrite 函数的功能是将声音数据按 μ 律压缩编码的方式存入声音文件中，其用法是：

》auwrite (y, file);　　y 是输入的声音信号，file 为字符串，它表示声音文件的文件名。

·auread 函数

auread 函数的功能是从 μ 律压缩编码方式的声音文件中读取声音数据，其用法是：

》 y = auread (file);　　　　y是输出的声音信号，file 为字符串，它表示声音 文件的文件名。

例 1　乐音信号发生器

audio 是一个用于产生乐音信号的函数，通过这个函数的学习和模仿，读者会对声音处理会有一个全面深入的了解，在对音频设备进行仿真的过程中可以借用其中的一些方法。

audio 函数有两个输入参数，一个是乐曲的速度 tempo，它表示每分钟多少拍；另一个是声音量化的比特数 nbits，这个参数仅适用于 5.x 版中。在函数内部首先确定了采样频率，并建立了 1 拍、2 拍和 3 拍的时间序列，接着定义了若干音符，每个音符采用了幅度随时间衰减的正弦波，然后将音符排列起来就生成了声音向量。

在程序中使用了平均律，即半音的频率比为 $2^{1/12}$，全音的频率比为 $2^{1/6}$，八度音的频率比为 2。我们知道 a1（即简谱中的 6）的频率是 440Hz，然后以它的频率为基准就不难求出其它各音的频率，如 c2（高音 C，简谱中的 i）与 a1 相差 3 个半音，故它们的频率比为 $2^{1/4} = 1.1892$，因此 c2 的频率应为 $440 \times 1.1892 = 523.25$ Hz。

由于声音处理在 4.2 版和 5.x 版之间存在着较大的差别，因此需要在程序中对版本进行判断，以便正确地使用声音处理函数。例如在 4.2 版中，为了存储声音文件，声音数据采用了 8 比特量化的方式。

程序清单如下：

```
function [] =audio (tempo, nbit);
%
%   Usage: audio (tempo);
if nargin<1; tempo=140; end;                        % 缺省值
if nargin<2; nbit=8;    end;
v=version;
if strcmp (v (1),'5');
  Ns= ['stereo, ' int2str (nbit) ' bits'] ;
else;
  Ns='mono';
end;
Ns= ['Musetto (a french song)                      BBI 2000' blanks (20) Ns];
figure ('Name', Ns,'Num','off');                   % 打开图形窗口
fa=440;    fs=fa*10;                               % 乐音的频率、采样频率
fb=493.88;    fc=523.25;
fd=587.33;    fe=659.26   ff=698.45;
T=60/tempo; dt=1/fs; t1=0: dt: T-dt; n=length (t1);    % 时间与拍子
t2=linspace (0, 2*T, 2*n); t3=linspace (0, 3*T, 3*n);
La=sin (pi*fa*t1) .*exp (-t1);                      % 伴奏的音符
La3=sin (pi*fa*t3) .*exp (-3*t3);
```

```
Do= .6 * sin (pi * fc * t1) .* exp (-t1);
Re= .6 * sin (pi * fd * t1) .* exp (-t1);
Mi= .6 * sin (pi * fe * t1) .* exp (-t1);
Fa= .6 * sin (pi * ff * t1) .* exp (-t1);
                                                  %旋律的音符
la= sin (2 * pi * fa * t1) .* exp (-t1);    la3= sin (2 * pi * fa * t3) .* exp (-t3);
si= sin (2 * pi * fb * t1) .* exp (-t1);
do= sin (2 * pi * fc * t1) .* exp (-t1);
do2= sin (2 * pi * fc * t2) .* exp (-t2); do3= sin (2 * pi * fc * t3) .* exp (-t3);
re= sin (2 * pi * fd * t1) .* exp (-t1);
mi= sin (2 * pi * fe * t1) .* exp (-t1); mi2= sin (2 * pi * fe * t2) .* exp (-t2);
fa= sin (2 * pi * ff * t1) .* exp (-t1);
y1= [la la la do2 do mi2 re do3 re mi re do2 si la3];
y1= [y1 la3 * exp (-3 * t3)]; t= dt * (0: length (y1) -1);
y2= [La Do Mi];    y2= [y2 y2 y2 y2 La Re Fa La Re Mi y2 La3];    % 和声
if strcmp (v (1),'5');                             % MATLAB 5.x
  ym= max (abs (y1));    y1= y1/ym;                % 右声道 (旋律)
  ym= max (abs (y2));    y2= y2/ym;                % 左声道 (和声)
  y= [y1' .75 * y2'];    sound (y, fs, nbit);
  wavwrite (y * .99, fs, nbit,'d: \audio');        % 储存声音文件
  ubplot (211), plot (t, y1,'b');
  title ('right channel');    grid; zoom xon;
  subplot (212), plot (t, y2,'m');    zoom xon;    grid;
  title ('left channel');    xlabel ('t (s)');
elseif strcmp (v (1),'4');                         % MATLAB 4.2
  y= y1 + .75 * y2;    ym= min (y);    y= y - ym;    % 单声道 (旋律 + 和声)
  ym= max (y);    k= fix (y/ym * 255);             % 8 比特量化
  sound (k, fs); whitebg ('w');
  wavwrite (k, fs,'d: \audio');                    % 储存声音文件
plot (t, k,'b'); grid; zoom xon; xlabel ('t (s)');
end;
```

3.3 图象处理

在 MATLAB 中专门设置了用于图象处理的函数，可以完成画图、动画等任务，同时可以从图形文件输入图象，也可以把图象存入图形文件中去。

3.3.1 建立图象

建立图象实际上就是画图（画画），注意此处所说的画图不是指绘制函数曲线，而是在

屏幕上画一幅图象。在MATLAB中，建立一幅图象要进行两部分的工作，首先是将屏幕划分成m×n个小方块，就像矩阵那样，屏幕上共有m行n列个小方块，每个方块就是一个象素；然后指定每个象素的颜色，这样就完成了一幅图象。

MATLAB给使用者准备了一些颜色库，称为颜色映象表（color map），它的作用就象是小学生用的水彩颜料一样，使用者可以在中间选择所需的颜色。所谓的颜色映象表实际上就是一个n×3的矩阵，其中每行的三个元素分别为代表红、绿、蓝，因此一行就是一种颜色，n行就定义了n种颜色。这样，颜色映象表也可称作是颜色矩阵，该矩形的行下标就代表着一种特定的颜色。因此画图的过程就是建立一个矩阵，该矩阵整体上代表着屏幕上的各个象素，矩阵的每个元素为一个整数，用它来指定颜色矩阵中的某一行，也就是说它的数值代表了颜色矩阵的行下标，这样就指定了该象素的颜色。除了现成的颜色映象表（矩阵）以外，使用者还可以建立自己的颜色矩阵。

· image 函数

image函数的功能是建立一个图象对象（即建立一幅图象），并返回一个句柄，其用法是：

》H=image（X）；　　X是个象素矩阵，下标则代表着屏幕上的各象素的位置，其元素的数值为整数，它是象素颜色的“索引”，H为图象对象的句柄。

》H=image（X，p1，v1，p2，v2，p3，v3，…）；　　函数的功能同上，同时对图象对象的若干属性进行设定。

image函数应该与colormap函数一起使用来确定象素的颜色，否则将按照缺省的颜色映象表来选择颜色。

· colormap 函数

colormap函数的功能是选择当前使用的颜色映象表，其用法是：

》colormap（map）；　　定义map（一个颜色矩阵）为当前使用的颜色映象表。

在MATLAB内部有一幅专门用于演示用的图象，其内容是个小丑，它的数据储存在数据文件clown.mat中。数据文件中储存的变量为2个：X和map，采用以下的程序就可将图象显示在屏幕上。

```
》load clown；  whitebg（'w'）；                        % 读数据文件，设定白色背景
》image（X）；  colormap（map）；  axis image；         % 画图，规定象素为正方形
》set（gca,'XTick'，[],'YTick'，[]）；                  % 去除坐标轴上的刻度
```

例2　电视的彩条图象

彩条信号是电视使用的一种标准信号，从左到右依次是白、黄、青、绿、品红、红、蓝、黑等八种颜色的竖条。通过模拟彩条图象，读者可以熟练地掌握MATLAB中的画图方式。

```
》X=1：8；  whitebg（'k'）；                    % 建立1×8象素向量，设定黑色背景
》map=［1 1 1；1 1 0；0 1 1；0 1 0］；          % 建立8种颜色矩阵
》map=［map；1 0 1；1 0 0；0 0 1；0 0 0］；
```

```
》image (X);   colormap (map);                      % 画图
》set (gca,'XTick', [],'YTick', []);                % 去除坐标轴上的刻度
》title ('COLOR BAR');
```

例3 电视的棋盘格图象(12×16个黑白格)

```
》m2=6;   n=16;   map= [0 0 0; 1 1 1];              %建立2种颜色矩阵
》k=rem (1: n, 2) +1;   kk=fliplr (k);              %定义象素矩阵中相邻的2行
》X= [];   whitebg ('k');
》for j=1: m2; X= [X; k; kk]; end;                  % 建立12×16象素矩阵
》image (X);   colormap (map); axis image;          %画图，规定象素为正方形
》set (gca,'XTick', [],'YTick', []);                % 去除坐标轴上的刻度
》title ('CHECKERBOARD');
```

3.3.2 颜色映象表与颜色函数

·颜色映象表

在MATLAB中设置了若干颜色映象表，在缺省的情况下，大多数的颜色映象表中有64种颜色，它们是：

hsv 色调-饱和度-强度颜色映象表

色调(H)、饱和度(S)、强度(V)是计算机中常用的表示颜色的方式，它与电视中使用的红绿蓝三基色之间存在着一定的换算关系。此映象表中设定了从红开始，下面依次是黄、绿、青、蓝、品红，最后又回到红。

》hsv (m);　　　共m种颜色。

gray　线性变化的灰度映象表

hot　黑-红-黄-白四色颜色映象表

cool　蓝-品红两色颜色映象表

bone　以蓝色为基调的颜色映象表

copper　以青铜为基调的颜色映象表

pink　以浅棕色为基调的颜色映象表

prism　共红、棕、黄、绿、蓝、紫六种颜色

jet　hsv颜色映象表的变形

flag　共红、白、蓝、黑四种颜色

·hsv2rgb函数

hsv2rgb函数的功能是将色调-饱和度-强度向量(矩阵)转换成红-绿-蓝向量(矩阵)，其用法是：

》[r g b] =hsv2rgb ([h s v]);　　　其中各参数为数量或列向量。

·rgb2hsv函数

rgb2hsv函数的功能是将红-绿-蓝向量(矩阵)转换成色调-饱和度-强度向量(矩阵)，其用法是：

》[h s v] =rgb2hsv ([r g h]);　　　其中各参数为数量或列向量。

· rgbplot 函数

rgbplot 函数的功能是将红、绿、蓝三条曲线将颜色映象表中的各种颜色的红－绿－蓝数值绘制出来，其横坐标为颜色矩阵的列下标，其用法是：

》rgb2plot (map);

·brighten 函数

brighten 函数的功能是改变颜色映象表中各种颜色的亮度，其用法是：

》hrighten (h);　　　0<h≤1 时，提高当前颜色映象表中各颜色的亮度，
　　　　　　　　　　−1≤ b< 0 时，降低当前颜色映象表中各颜色的亮度，

》map=hrighten (b);　　　将亮度改变后的颜色映象表定义为 map。

》map1=brighten (map, b);　将 map 的亮度改变，生成新的颜色映象表 map1。

3.3.3 图象对象的属性

在 MATLAB 中，image 函数建立的对象为图象对象，图象对象是坐标对象的子对象，是最底层的对象。

》H=image (x); colormap (map); get (H);　便获取了当前图象对象的所有属性。不同版本的 MATLAB 的图象对象的属性是不同的，4.2 版本的典型结果为：

CData = [(1 hy 100)]

XData = [1 100]

YData = [1]

ButtonDownFcn =

Children = []

Clipping = on

Interruptihle = no

Parent = [113.001]

Type = image

UserData = []

Visible = on

3.3.4 图象的输入与输出

在 5.x 版中的 MATLAB 中设有图象输入和输出的函数，而在 4.2 版中，图象输入和输入函数是放在 IMAGE 工具箱内的，并且功能也与 5.x 版的不同。此处只介绍 5.x 版中的图象输入和输出函数。

· imread 函数

imread 函数的功能是从一个图形文件中读取图象的数据，其基本用法是：

》[X, map] =imread (file); 输出参数 X 为象素矩阵，map 为颜色映象表。file 为图形文件的文件名，它是个字符串。文件名应该包括后缀，通过后缀来判断图形文件的格式。

imread 函数支持几种图形格式，它们是：

1) BMP 格式　　1 比特、4 比特、8 比特、24 比特未压缩的图象，4 比特、8 比特采用游程编码 (RLE) 的图象。

2) HDF 格式　　8 比特、24 比特栅状数据格式。

3) JPEG 格式　通常的 JPEG 格式。

4) PCX 格式　1 比特、8 比特、24 比特的图象。

5) TIFF 格式　1 比特、8 比特、24 比特未压缩的图象，1 比特、8 比特、24 比特采用 packbits 压缩方式的图象，1 比特采用 CCITT 压缩方式的图象。不支持 LZW 压缩方式。

6) XWD 格式　1 比特、8 比特的图象。

· imwrite 函数

imwrite 函数的功能是将图象数据写入一个图形文件中，其基本用法是：

》imwrite（X，map，file）；　输入参数 X 为象素矩阵，map 为颜色映象表。file 为图形文件的文件名，它是个字符串。文件名应该包括后缀，通过后缀来判断图形文件的格式。

imwrite 函数支持几种图形格式，它们是：BMP、HDF、JPEG（JPG）、PCX、TIFF、XWD。

3.4　三维图形

在 MATLAB 中提供了若干用于绘制和处理三维图形的函数。需要指出的是，在三维图形方面，4.2 版和 5.x 版之间存在着较大的差异。三维图形包括三维曲线、网格图、曲面图、等高线图等不同的类型。

3.4.1　三维图形的投影

我们知道，计算机屏幕和纸张都是二维的，因此在上面只能显示或绘制平面图形（二维图形）；要想在它们上面反映出立体图形（三维图形），就必须将三维图形投影到一个平面上，也就是说要进行三维与二维之间变换。

通常投影的方式有两种，一种是正投影（orthographic projection），即使用平行光线进行投影；另一种为中心投影或透视投影（perspective projection），即使用点光源进行投影。正投影多用于制图，它相对简单一些，面透视投影比较复杂，但与人眼观察的效果相当接近。

·投影与变换矩阵

进行三维图形与二维图形的变换，使用矩阵的方式是比较清楚的，同时表达方式也比较简单。坐标变换的方式有比例、旋转、错切、平移、透视等等，设三维空间中一个点在直角坐标系中的坐标是［x y z］，而在另一个正交坐标系中的坐标是［u v w］，面［u v］就代表了投影面，将［x y z］变换成［u v］就完成了三维与二维间的投影变换。

考虑到透视和平移，一般的变换关系可以表示为：$\begin{bmatrix} H_u \\ H_v \\ H_w \\ H \end{bmatrix} = \begin{bmatrix} a & d & g & l \\ b & e & h & m \\ c & f & i & m \\ p & q & r & s \end{bmatrix}$，其中 4×4 的方阵就称为变换矩阵，记为 T，

而 $u = H_u/H$，$v = H_v/H$，$w = H_w/H$。

在变换矩阵T中，$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$代表着比例、旋转、反射、错切变换，$\begin{bmatrix} l \\ m \\ n \end{bmatrix}$代表着平移变换，$[p \quad q \quad r]$代表着透视变换，$s$为全比例因子。变换矩阵中各元素的数值由光的入射角度（观察者的视角）和光源的位置（观察者的位置）来确定。

视角有两个，一个是方位角，记为az，另一个是仰角，记为el。在MATLAB中，方位角是在xy平面上从$-y$轴开始向右计算的，即$az=0°$对应着$-y$轴，而$az=90°$对应着x轴；el称为仰角，在MATLAB中仰角是从xy平面向上计算的，即$el=0°$对应着xy平面，而$el=90°$对应着$+z$轴。

对于正投影，变换矩阵可表示为

$$[T]=\begin{bmatrix} \cos az & \sin az & 0 & 0 \\ -\sin el \cdot \sin az & \sin el \cdot \cos az & \cos el & 0 \\ \cos el \cdot \sin az & -\cos el \cdot \cos az & \sin el & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

对于透视投影，变换矩阵可表示为

$$[T]=\begin{bmatrix} \cos az & \sin az & 0 & 0 \\ -\sin el \cdot \sin az & \sin el \cdot \cos az & \cos el & 0 \\ \cos el \cdot \sin az & -\cos el \cdot \cos az & \sin el & 0 \\ -\cos el \cdot \sin az/f & \cos el \cdot \cos az/f & -\sin el/f & 1 \end{bmatrix}$$

在制图中，经常使用的投影方式有，正等侧投影和正二侧投影。正等侧投影时，方位角az=135°，仰角el=35.3°，确保了三个轴的长度相同；面正二侧投影时，方位角az=110.7°，仰角el=19.5°，确保了两个轴的长度相同。

·view函数

view函数的基本功能是改变视角。在MATLAB中，视角的缺省值为：az = −37.5°，el = 30°。view函数的用法是：

〉view (az, el); 改变视角（方位角az和仰角el）。

〉view ([x y z]); 用矢量来表示投影光线的方向。

〉T=view; 输出当前的变换矩阵T。

〉view (T); 采用变换矩阵T来投影。

·viewmtx函数

viewmtx函数的功能是计算变换矩阵T，其基本用法是：

〉T=viewmtx (az, el); 根据方位角az和仰角el，计算正投影变换矩阵T。

〉T=viewmtx (az, el, phi); 根据方位角az、仰角el和对视角phi，计算透视变换矩阵T。上述三个参数的单位都是度，其中对视角与透视变换矩阵中的参数f的关系为：phi=atan (1/f/sqrt (2)) *360/pi

在4.2版中，显示透视投影的效果可以采用以下的方法：先设定方位角az、仰角el和对视角phi，便得到一个透视投影变换矩阵，然后使用它来进行观察，见下面的程序段，其

中 cylinder 是用来显示柱面的 MATLAB 函数。

》cylinder; T=viewmtx (45, 60, 66); view (T);

在 5.x 版中实施透视投影是很简单的，首先对投影的方式进行说明，同时还应指定投影中心的位置，具体方法如下：

》cylinder; camproj ('pers'); campos ([4 4 3]);

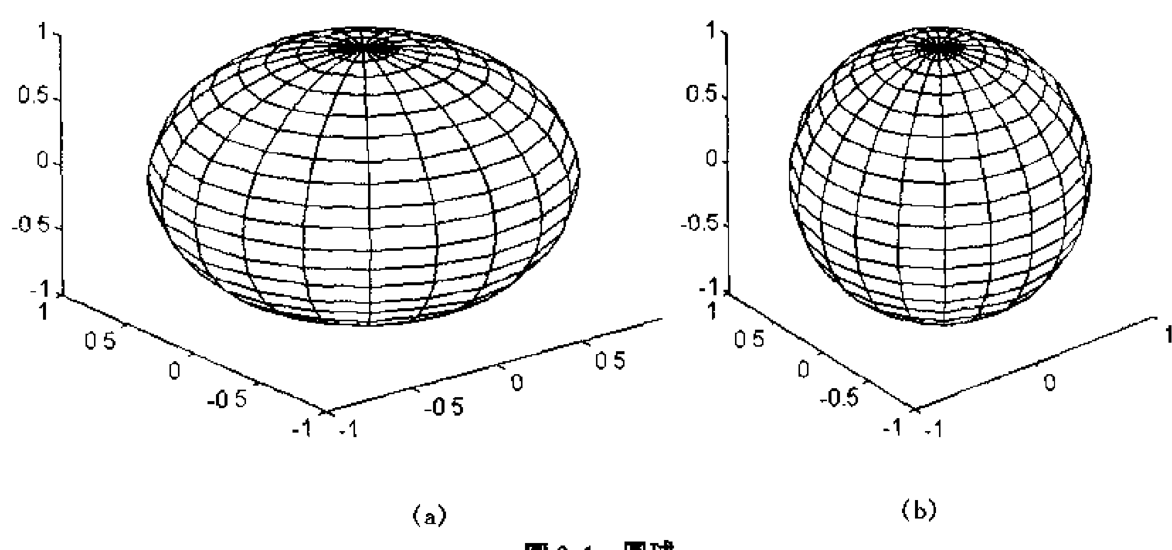
其中 camproj 函数用来指定投影的方式，透视投影可简化为 pers；而 campos 函数用位置向量来指定投影中心的位置。

·三维图形显示格式的处理

由于 MATLAB 在绘制图形时充分利用屏幕的尺寸，由于屏幕的宽高比为 4：3，因此投影后，图形的水平分量被夸张了，这在显示圆球时特别明显，若不加处理的话，圆球就变成了椭球，在这点上与在平面上绘制圆形的情况是类似的。

MATLAB 中有一个专门用来显示圆球的函数，为 sphere 函数，在 4.2 版和 5.x 版中直接使用 sphere 函数，画出的图形见图 3.1a，显然圆球变成了扁椭球。

》[x, y, z] =sphere; whitebg ('w'); mesh (x, y, z); colormap ([0 0 0]);



(a)　　　　(b)

图 3.1 圆球

为了正确地显示投影的结果，必须对坐标对象的属性进行正确的设置。对于 4.2 版来说，坐标对象中有个 AspectRatio（图象尺寸比）属性，它是用来控制图象的宽高比和坐标轴的数据单位，合适地定义宽高比，就可以准确地显示三维图形。

对于正投影来说，宽高比就是 T 矩阵第一行的元素和与第二行的元素和之比的绝对值，于是有：

》[x, y, z] =sphere; whitebg ('w'); mesh (x, y, z); colormap ([0 0 0]);

》T=view; w=sum (ahs (T (1,:))); h=sum (abs (T (2,:)));

》set (gca,'asp', [w/h NaN]);

显示结果见图 3.1b。

对于 5.x 版来说，坐标对象中有个 PlotBoxAspectRatio（图象尺寸比）属性，它是用来控制图象的宽高比和坐标轴的数据单位。采用下面的方法就可以正确地显示三维图形，

```
》[x, y, z] =sphere; mesh (x, y, z); colormap ( [0 0 0]);
》set (gca,'plotbox', [1 1 1]);
```

上述的两种三维图形显示格式的处理方法适用于各种视角的正投影。

3.4.2 plot3 函数

plot3 函数用来绘制三维曲线，它将 plot 函数扩展到三维空间，因此其使用方法与 plot 函数相近。plot3 函数的基本用法是：

》H=plot (x, y, z, c);	根据向量 x、y 和 z 绘制三维折线，字符串 c 用来表示折线的类型和颜色，返回线条对象句柄 H。
》H=plot (x, y, z);	根据缺省的折线的类型和颜色绘制三维折线。
》H=plot (X, Y, Z, c);	根据矩阵 X、Y 和 Z 的列向量绘制多根三维折线。

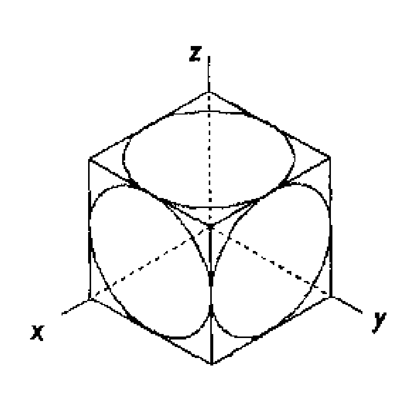
例 4 正等轴侧图和正二轴侧图

介绍这个例子的目的是为了使读者对三维绘图有个全面的和透彻的了解。函数的程序清单如下：

```
function [] =draft ();
%                          BBI 1999
v=version;   AZ= [45 20.7] +90;   EL= [35.3 19.5];
Nstr='Isometric and Dimetric        BBI 99/10';
figure ('Name', Nstr,'Num','off','Pos', [5 35 790 520]);
t=0: 360; t=t' * pi/180;          E2=ones (size (t)) *2;
sint1=sin (t) +1;                 cost1=cos (t) +1;
whitehg ('w');                    dx= [.3 .9];
for i=1: 2;
  subplot (1, 2, i);
  plot3 ( [sint1 sint1 E2], [cost1 E2 sint1], [E2 cost1 cost1],'b');
  hold on; axis off;      hidden off;
  view (AZ (i), EL (i));       T=view;
  x= [0 0 0; 2 0 0]; y= [0 0 0; 0 2 0]; z= [0 0 0; 0 0 2];
  x1= [2.5 0 0; 2 0 0]; y1= [0 2.5 0; 0 2 0]; z1= [0 0 2.5; 0 0 2];
  plot3 (x, y, z,'k:', x1, y1, z1,'k');
  x= [0 0 0; 2 2 2]; y= [0 2 2; 0 2 2]; z= [2 0 2; 2 0 2];
  plot3 (x, y, z,'k', y, z, x,'k', z, x, y,'k');
  if strcmp (v (1),'4');
    w=sum (ahs (T (1,:)));      h=sum (abs (T (2,:)));
    set (gca,'AspectRatio', [w/h NaN]); %MATLAB 4.x
  elseif strcmp (v (1),'5');
    set (gca,'PlotBoxAspectRatio', [1 1 1],'Box','off');
    set (gcf,'Color', [1 1 1]);             %MATLAB 5.x
```

```
    end;
    H=text (3, 0, 0,'x','FontSize', 14,'FontA','italic');
    H=text (0, 2.7, 0,'y','FontSize', 14,'FontA','italic');
    H=text (dx (i), 0, 2.7,'z','FontSize', 14,'FontA','italic');
end;            hold off;
```

draft 函数的运行结果如图 3.2 所示。



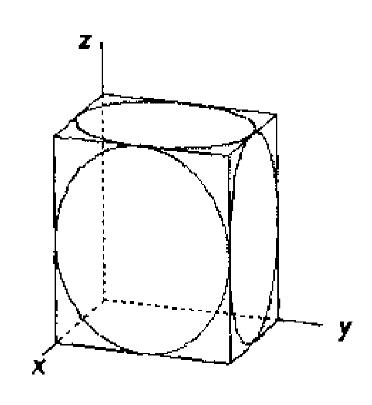


图 3.2　正等轴侧图和正二轴侧图

3.4.3　网格图和曲面图

plot3 函数用来绘制三维空间内的曲线，它生成的是线条对象，而网格图和曲面图表示的是三维空间内的曲面，它们生成的是曲面对象。在 MATLAB 中，三维空间内的曲面是用网格或网格加颜色来表示的，这里的颜色可以是模拟的自然光，也可以是一种表示高度的特殊坐标（称为伪彩色）。

我们知道，二元函数 z=f (x, y) 的几何意义就是三维空间内的曲面，其中 z 表示着曲而在 xy 平面上的高度，x 和 y 为两个自变量。

·mesh 函数

mesh 函数是一个基本的三维绘图函数，它的功能是绘制网格图，并用伪彩色来表示曲面的高度，其基本用法是：

〉H=mesh (x, y, z, c);　　输入参数 x 和 y 均为矩阵，它们表示着曲而存在的区域，z 矩阵表明的是曲面的高度，函数根据字符串 c 表示的颜色映象表来绘制网格图。输出参数 H 为曲面对象的句柄。

〉H=mesh (x, y, z);　　根据缺省的颜色映象表绘制网格图。

〉H=mesh (z, c);　　用 z 矩阵的下标替代 x 和 y 矩阵。

·几个曲面生成函数

MATLAB 提供了几个现成的曲面生成函数，如：sphere、cylinder、peaks。

〉[x, y, z] =sphere (n);　　生成一个球而。输入参数 n 用来控制输出参数矩

	阵的大小，缺省值为：n=20。
》[x，y，z]=cylinder（r，n）；	生成一个圆柱面或圆锥面。输入参数 n 用来控制输出参数矩阵的大小，缺省值为：n = 20。r=[r1 r2]，分别代表上下两个半径，缺省值为：r=[1 1]。圆柱面或圆锥面的高度为 1。
》[x，y，z]=peaks（n）；	生成一个类似于山峰和山谷的曲面。输出参数均为 n×n 矩阵，缺省值为：n=49。

·hidden 函数

hidden 函数是来控制网格图的消隐方式，其用法是：

》hidden on；	对网格图设置消隐（不透明的网格图）。
》hidden off；	对网格图不设置消隐（透明的网格图）。

·surf 函数

surf 函数是一个基本的三维绘图函数，它的功能是绘制曲面图，并用伪彩色来表示曲面的高度，其基本用法是：

》H=surf（x，y，z，c）；	输入参数 x 和 y 均为矩阵，它们表示着曲面存在的区 域，z 矩阵表明的是曲面的高度，函数根据字符串 c 表示的颜色映象表来绘制曲面图，曲面上既有网格，也有颜色。输出参数 H 为曲面对象的句柄。
》H=surf（x，y，z）；	根据缺省的颜色映象表绘制曲面图。
》H=surf（z，c）；	用 z 矩阵的下标替代 x 和 y 矩阵。

· shading 函数

shading 函数用来控制曲面对象和补片对象的彩色阴影方式，其用法是：

》shading flat；	在面元上使用同一颜色。
》shading interp；	对面元上的颜色进行插值。
》shading faceted；	设置表面阴影（缺省值）。

· 矩形区域的确定

在绘制网格图和曲面图时，必然要涉及到图形的区域问题，$a<x<h$ 和 $c<y<d$ 的区域为矩形区域。对于矩形区域，MATLAB 专门设置了一个 meshgrid 函数，用来计算 x 矩阵和 y 矩阵。如：

》[x，y]=meshgrid（-2：.1：4，0：.1：3）；　表示 $-2\leqslant x\leqslant 4$ 和 $0\leqslant y\leqslant 3$ 这样一个区域。

· 圆形区域的确定

对于圆形区域，读者可以使用下面的自定义函数 meshcirc。

》[x, y] =meshcirc (r, n);　　表示半径为 r 的圆形区域，x 和 y 为 n×n 矩阵。

```
function [x, y] =meshcirc (r, n);
%
% BBI 2000
if nargin<1; r=1; end;                          % 缺省值
if nargin<2; n=17; end;
dr=r/ (n-1);    rh=0: dr: r;    rh=rh';         % 列向量
dth=360/ (n-1);    th=0: dth: 360;              % 行向量
sint=sin (th * pi/180); cost=cos (th * pi/180);
x=rh * cost; y=rh * sint;                       % 方阵
```

·meshz 函数

meshz 函数是 mesh 函数的基础上添加上了参考平面，其用法是：

》H=meshz (x, y, z, c);

·surfl 函数

surfl 函数也是用来绘制曲面图的，它与 surf 函数的区别在于颜色方面。surfl 函数使用自然光来表示曲面，其效果与人眼观察到的相近，其基本用法是：

》H=surfl (x, y, z, s);　　输入参数 x 和 y 均为矩阵，它们表示着曲面存在的区域，z 矩阵表明的是曲面的高度。s 参数用来表示光源的方向，s= [az el]，合理地使用光源完善曲面的效果。输出参数 H 为曲面对象的句柄。

》H=surfl (x, y, z);　　根据缺省的光源来绘制曲面图，此时光源位于观察 方向左侧 45°的方向上。

例 5　网格图与曲面图

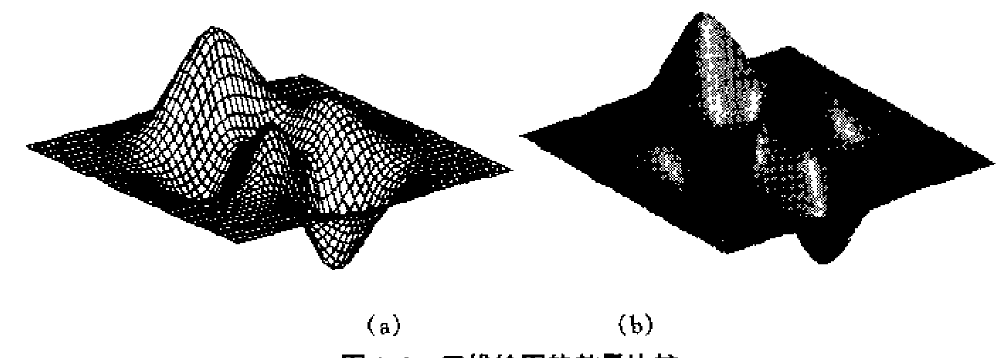
通过这个例子，读者可以对网格图与曲面图的异同之处进行比较，见图 3.3。

```
》whitebg ('w'); mesh (peaks); colormap ( [0 0 0]);    % 网格图
》whitebg ('w'); surfl (peaks (200)); colormap (bone);    % 曲面图
》shading interp; axis image; axis off; brighten (.1);
```

·fill3 函数

fill3 函数用来绘制三维空间内的多边形，它将 fill 函数扩展到三维空间，因此其使用方法与 fill 函数相近。fill 函数和 fill3 函数生成的对象称为补片对象。fill3 函数的基本用法是：

》H=fill3 (x, y, z, c);　　输入参数 x、y 和 z 为向量，表示着一个多边形的几何坐标；若输入参数为矩阵，则每个列向量表示一个多边形。字符串 c 表示着多边形的颜色。



(a)　　　　　　(b)

图3.3　三维绘图的效果比较

3.4.4　等高线图

等高线图是用高度一定的水平平面与三维空间内曲面相交而得到的，同时类似的方法可以用来绘制二元函数的曲线族。

·contour 函数

contour 函数生成等高线图，其基本用法是：

》[a, H] =contour (x, y, z, v, c)	矩阵 x、y 和 z 用来表示空间的曲面，向量 v 表示等高线的数值，字符串 c 用来指定曲线的颜色和类型。a 是 clabel 函数输入参数，两函数联合使用可以在图上对等高线进行说明。H 是 句柄，此函数生成线条对象。
》[a, H] =contour (x, y, z);	使用缺省的数值及缺省的曲线颜色和类型。
》[a, H] =contour (z, v, c);	使用 z 矩阵的下标替代 x 和 y 坐标。

· clabel 函数

clabel 函数与 contour 函数一同使用，用来在等高线上加上高度数值，其用法是：

》clabel (a);	a 为 contour 函数的输出参数。

·mesbc 函数

meshc 函数的功能是进行两格和等高线混和绘图。

·surfc 函数

surfc 函数的功能是进行曲面和等高线混和绘图。

·contour3 函数

contour3 函数生成三维的等高线图。

例 6　绘制二元函数的曲线族

在工程问题中，经常会遇到二元函数的问题，采用 contour 函数可以较好地满足工程曲线的绘制任务。

如卫星接收天线的仰角 EL 与接收地点的经度 ξ 和纬度 λ 有关，同时与卫星的位置 λs

有关，其表达式为：

$$\tan EL = \frac{\cos\xi\cos(\lambda - \lambda_s) - 0.15127}{\sqrt{1 - \cos^2\xi\cos^2(\lambda - \lambda_s)}}$$

显然，函数的自变量有两个，纬度 ξ 和相对经度 $\lambda - \lambda s$。下面的自定义函数就是用来绘制天线仰角的曲线图，如图 3.4 所示。程序清单如下：

```
function [xi, lmd, el] =elsat ();
%      BBI 2000
[lmd, xi] =meshgrid (0: 90, 0: 90);                    %生成自变量矩阵
cosd=cos (xi * pi/180) .* cos (lmd * pi/180);          % 计算公式
tanel= (cosd-0.15127) ./sqrt (1-cosd.^2+ (cosd==1) * eps);
el=atan (tanel) * 180/pi;
[sc, H] =contour (lmd, xi, el, 0: 10: 90,'k');         %等高线图
clabel (sc);      grid; whitebg ('w');
axis ( [0 90 0 90]); axis image;
set (gca,'xtick', 0: 10: 90);
xlabel ('l');       Hx=get (gca,'xlabel');
set (Hx,'FontN','symbol','FontSize', 14);              %定义希腊字母
ylabel ('x');       Hy=get (gca,'ylabel');
set (Hy,'FontN','symbol','FontSize', 14);
```

由于在 MATLAB 中可将向量和矩阵当作“数”来处理，因此大大地简化程序的结构，使程序本身容易被读懂，同时由于避免了使用多重循环，故加快了程序的运行速度。从这个自定义函数中，可清楚地看到 MATLAB 语言的特点。

3.4.5 三维动画

MATLAB 中设置了三个函数用来处理动画问题，它可以存储一系列的二维或三维图象，然后像放电影一样把它们按次序重放出来。运行动画最好在 5.x 版的 MATLAB 中进行。

· moviein 函数

moviein 函数用来为存储动画画面面对内存进行初始化，其用法是：

》M=moviein (n);　　　M 是一个充分大的矩阵，它用来储存 n 帧画面，每列储存一帧。

· getframe 函数

getframe 函数用来获取当前屏幕上的一帧画面，输出为列向量。

· movie 函数

movie 函数用来运行动画画面，其用法是：

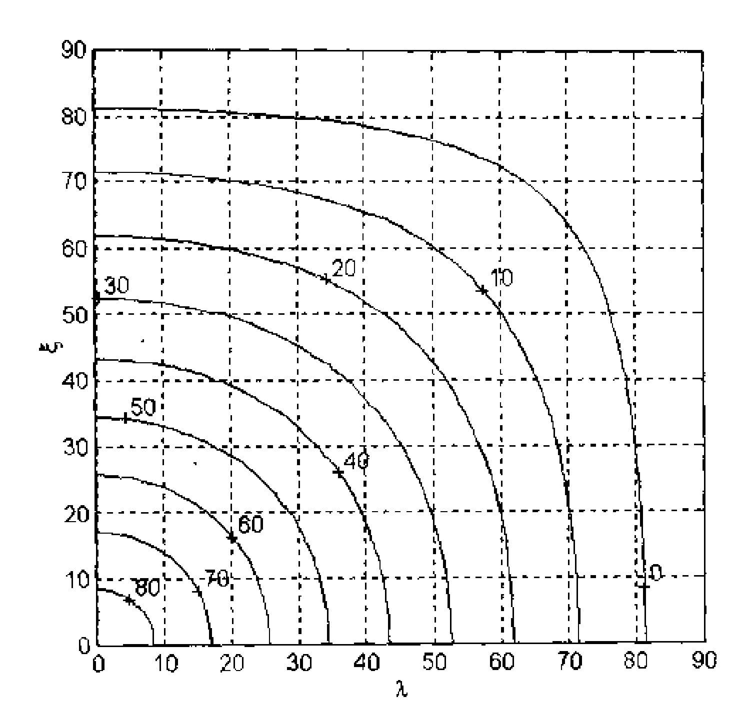


图 3.4 卫星接收天线的仰角曲线

》movie (M, n);　　运行 M 中储存的画面，共运行 n 次。

》movie (M);　　运行 M 中储存的画面 1 次。

例 7　沿 z 轴旋转的三维动画

通过这个例子来具体说明动画的制作方式。首先建立一帧静止的三维画面，然后调用一个自定义函数 mmspin3d 来完成画面的旋转，方法如下：

》[x, y, z] =peaks (30); surfl (x, y, z); shading interp; colormap (bone);

》mmspin3d;

自定义函数 mmspin3d 的程序清单如下：

```
function M=mmspin3d (n);
% Copyritht © Prentice-Hall, Inc.
if nargin<1; n=18; end;
n=max (abs (round (n)), 2); axis (axis); axis off;
incaz=round (360/n);        [az, el] =view;
title ('Input data, wait please.');
m=moviein (n);
for i=1: n;
   view (az+incaz* (i-1), el); m (:, i) =getframe;
end;
H=get (gca,'title');
set (H,'string','Now, the movie begins.'); pause (1);
if nargout;
```

```
  M=m;
else;
  movie (m, 3);
end;
set (H,'string','END','fonta','italic');
```

3.5 菜单式界面

使用 MATLAB 可以建立起类似于微软视窗操作系统中菜单栏内的标准菜单，于是在很多情况下，可以使用一种非常简单的方式来对 MATLAB 的各个函数进行操作了。这样就建立起一种交互式的界面，使用鼠标就可以完成很多工作了。显然，这样一来使 MATLAB 函数看上去类似于一种应用软件，故便于推广 MATLAB 的应用范围。同时，建立自定义的菜单式界面并不复杂，读者可以很快地掌握。

3.5.1 用户菜单

使用 uimenu 函数就可以建立起用户菜单，用户菜单也是一种对象，称为用户菜单对象，它与坐标对象一样，属于第二层对象。

· uimenu 函数

uimenu 函数的功能就是建立用户菜单和用户子菜单，它们出现在视窗操作系统中的菜单栏上，与视窗的菜单形式上是完全一样的，其用法是：

》H=uimenu (p1, v1, p2, v2, p3, v3, p4, v4, …);

在图形窗口的菜单栏上建立起一个用户菜单，用户菜单依次出现在 help 菜单之后。p1, p2, p3, p4 等为用户菜单对象的属性，而 v1, v2, v3, v4 等则为相应的属性值，弄清楚各属性和属性值才能正确地使用用户菜单。H 为用户菜单的句柄。

》H2=uimenu (H1, p1, v1, p2, v2, p3, v3, p4, v4, …);

建立起一个用户子菜单，其中 H1 为上一级用户菜单的句柄。

4.2 版中用户菜单对象的各种属性如下：

Accelerator
BackgroundColor
CallBack
Checked: [on | {off}]
Enable: [on | {off}]
ForegroundColor
Label
Position
Separator: [on | {off}]
ButtonDownFcn
Clipping: [{on} | off]
Interruptible: [{no} | yes]
Parent
UserData
Visible: [{on} | off]

在各项对象属性中，Label 和 Callback 是最基本的，同时也是最重要的。

· Label（菜单名）

菜单名的属性值是一个字符串，它就是菜单上显示的名字。在字符串中的某个字符前面加上 & 符号，就定义了其后的那个字母为 PC 计算机中的快捷键（Alt－字母）。

属性名可简化为 Lab。

》H＝uimenu（'Label'，'&Play'）；　建立了一个菜单，名为 Play，p 为快捷键。

》H1＝uimenu（H，'Label'，'&Stereo'）；　在 Play 菜单下，建立了一个子菜单，名为 Stereo，s 为快捷键。

》H2＝uimenu（H，'Label'，'&Mono'）；　同样在 Play 菜单下，建立了一个子菜单，名为 Mono，m 为快捷键。

》H3＝uimenu（H，'Label'，'&Balance'）；　在 Play 菜单下建立了另一个子菜单。

》H11＝uimenu（H1，'Label'，'&Laft'）；　在 Stereo 菜单下建立了一个子菜单。

》H12＝uimenu（H1，'Label'，'&Right'）；　在 Stereo 菜单下建立了另一个子菜单。

以上的程序段的运行结果见图 3.5。

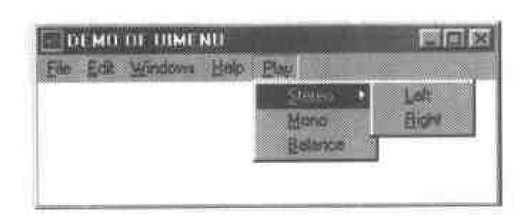


图 3.5　用户菜单界面

· Callback（回应）

回应代表着使用鼠标点击这个菜单后 MATLAB 应该执行的功能，回应的属性值为一字符串，其内容应该是 MATLAB 的命令、函数、程序段以及 MATLAB 表达式。在使用鼠标选中该菜单后，MATLAB 内部将使用 eval 函数，来执行该字符串所表示的的功能。属性名可简化为 Call。

需要进行说明的是，在函数 eval（s）中的输入参数 s 只能是 1×N 的字符串，而不能是字符串矩阵。例如：

》s＝str2mat（'plot（1：11）；'，'grid'）；eval（s）；是错误的，不能运行。应改为

》s＝'plot（1：11）；grid'；eval（s）；

在实际编程过程中，回应的字符串中往往会遇到必须使用单引号的问题。在字符串所表示的 MATLAB 表达式这中出现的单引号应该改为使用两个单引号。如在上面的例子中要使用蓝色画曲线时，应该输入 plot（1：11，'b'）；grid，此时就产生了引号的问题，应该采用以下的方式，

》s＝'plot（1：11）；''b'' grid'；eval（s）；　　其结果是准确无误的。

例如，菜单的功能就是要绘制上面出现曲线的话，那么就有：

》H＝uimenu（'Label'，'&Plot'，'Callback'，'plot（1：100，''m''）；grid；'）；

除了上述两项基本属性之外，用户菜单对象还有一些其它的属性，如：

· Checked（选中）

选中属性的属性值有 2 个：[on | {off}]，缺省值为 off。当属性值为 on 时，在菜单的左面会出现一个√标记。

· Enable（激活）

激活属性的属性值有 2 个：[{on} | off]，缺省值为 on。当属性值为 off 时，菜单的标记变灰，表示此菜单无效。属性名可简化为 En。

· Position（位置编号）

位置编号的属性值反映了菜单的位置，编号的次序在子菜单（下拉菜单）中是从上到下，在视窗的菜单栏中是从左到右。属性名可简化为 Pos。

· Separator（分割线）

与视窗类似，在下拉菜单内可以设置分割线，分割线的属性值有 2 个：[on | {off}]，缺省值为 off。当属性值为 on 时，在下拉菜单中当前菜单的上面就会出现一条分割线。属性名可简化为 Sep。

· UserData（用户数据）

用户数据提供了一个矩阵或一个字符串矩阵，该矩阵给用户菜单的句柄附加上一些相关的数据参数，合理地使用这个属性，可以非常灵活地在 M 函数和用户菜单之间进行数据的传递。属性名可以简化为 User。

```
》 H = uimenu ('Label', '&Plot', 'Call', …
            'z = get (H, ''user''); surfl (z); colormap bone; shading interp;');
```

上面所示的用户菜单的回应是画一帧曲面图，但是并未指定绘图函数的输入参数，在这种情况下显然是不能完成上述任务的，但是若添加上以下的一句程序，对用户数据进行定义，这样使用鼠标点击 Plot 菜单就可以完成曲面图的绘制工作了。

```
》 set (H, 'user', peaks);
```

3.5.2 菜单式界面的设计

使用菜单式界面的好处是，可以在同一个图形窗口的环境下灵活地选择不同的工作任务，也就是调用不同的 M 函数或执行不同的程序段。菜单式界面是十分友好的，特别是对于那些 MATLAB 应用程序的普通使用者（不是程序的编写者）来说，基本上不用学习 MATLAB，就可以方便地运行用 MATLAB 编写的程序，就类似于使用视窗操作系统一样，这就是菜单式界面明显优于普通命令窗口之处。

与按钮式界面（控制框界面）相比，菜单式界面的设计要简单得多。只需将可能使用的 M 函数或程序段按照其功能先进行分组，将功能类似或相互关联的用户菜单设置在一起就可以了，分组可以采用子菜单的方式，子菜单还可进一步分为一级子菜单、二级子菜单、三级子菜单等等。

在功能比较简单的情况下，菜单的回应可以采用程序段的方式，最好先定义一个回应字

符串，然后再建立用户菜单，这样程序的结构显得比较简明清楚，便于阅读，便于修改，便于更新；而在功能比较复杂的情况下，则应该调用相对应的 M 函数，这样一来建立菜单界面的程序段就变得非常简洁。

3.5.3　实例

从编程的角度来看，菜单的回应采用调用 M 函数的方式是非常简单的，而回应采用程序段的方式则相对复杂一些，故在此处介绍一个采用程序段作为菜单回应的例子，其功能是显示正投影和透视投影（中心投影）产生的不同效果。

```
function [] = projdemo ();
%
%  BBI 2000
figure ('Num','off','Menu','none','Name', ...
   'Demo of Projection      BBI 2000', ...
  'units','normal','pos', [.01 .03 .985 .9]);
v = version;
if strcmp (v (1),'5');        % MATLAB 5.x
  c='cylinder ( [1 .5], 36); colormap ( [1 .6 .3]);';
  c= [c 'set (gca,''plotbox'', [1 1 1],''zlim'', [0 2]);'];
  c11 = c;       c2 = 'set (gca,''proj'',''pers'');';
  c12= [c 'view (45, 35.3);'];    c13= [c 'view (20.7, 19.5);'];
  c14= [c 'view (90, 0);'];    c15= [c 'view (0, 90);'];
  c22='cylinder ( [1 .5], 36); ';
  c22= [c22 'set (gca,''plotbox'', [1 1 1],''proj'',''pers'');'];
  c22= [c22 'campos ( [0 0 2]);'];
elseif strcmp (v (1),'4');        % MATLAB 4.2
  c='cylinder ( [1 .5], 36); colormap ( [1 .6 .3]);';
  c12= [c 'view (45, 35.3);'];    c13= [c 'view (20.7, 19.5);'];
  c14= [c 'view (90, 0);'];    c15= [c 'view (0, 90);'];
  c42='T=view; w=sum (abs (T (1, 1: 3))); h=sum (abs (T (2, 1: 3)));';
  c42= [c42 'set (gca,''asp'', [w/h nan],''zlim'', [0 2]);'];
  c= [c c42];   c11= [c c42];    c12= [c12 c42];
  c13= [c13 c42];   c14= [c14 c42];    c15= [c15 c42];
  c2=' [az, el] =view; T=viewmtx (az, el, 25); view (T);';
  c2= [c2 c42]; whitehg ( [.8 .8 .8]);
  c22= [c 'T=viewmtx (0, 90, 38.94, [.5 .5 0]), view (T);'];
  c22= [c22 'w=sum (abs (T (1, 1: 3))); h=sum (ahs (T (2, 1: 3)));'];
  c22= [c22 'set (gca,''asp'', [w/h nan],''zlimmod'',''auto'');'];
end;
```

```
t1='title (''Orthographic'');'; t2='title (''Perspective'');';
H1=uimenu ('lab','&Orthographic');          %正投影菜单
  H11=uimenu (H1,'lab','&Normal','call', [c11 t1]);
  H12=uimenu (H1,'lab','&Isometric','call', [c12 t1]);
  H13=uimenu (H1,'lab','&Dimetric','call', [c13 t1]);
  H14=uimenu (H1,'lab','&X-axis','call', [c14 t1]);
  H15=uimenu (H1,'lab','&Z-axis','call', [c15 t1]);
H2=uimenu ('lab','&Perspective');          %透视投影菜单
  H21=uimenu (H2,'lab','&Normal','call', [c11 c2 t2]);
  H24=uimenu (H2,'lab','&X-axis','call', [c14 c2 t2]);
  H25=uimenu (H2,'lab','&Z-axis','call', [c15 c2 t2]);
  H22=uimenu (H2,'lab','&Top point','sep','on','call', [c22 t2]);
eval (c);
```

3.6 按钮式界面

使用 MATLAB 可以建立按钮、滑标、文本框、弹出式菜单等等控制框，从而建立起一种比用户菜单界面更为灵活的交互式界面，这种界面称为控制框界面或按钮式界面。按钮式界面的优点与用户菜单界面的优点类似，但是使用起来更方便，功能更强大，但是按钮式界面的编程要相对复杂一些。

3.6.1 用户控制框

使用 uicontrol 函数就可以建立起各种控制框，它也是一种对象，称为用户控制框对象，它与坐标对象一样，属于第二层对象。

· uicontrol 函数

uicontrol 函数的功能就是建立各种控制框，其用法是：

》H=uicontrol (p1, v1, p2, v2, p3, v3, p4, v4, …);

p1，p2，p3，p4 等为用户菜单对象的属性，而 v1，v2，v3，v4 等则为相应的属性值，弄清楚各属性和属性值才能正确地使用 uicontrol 函数。H 为用户菜单的句柄。

4.2 版中用户控制框对象的各种属性如下：

BackgroundColor

CallBack

ForegroundColor

HorizontalAlignment：[left | {center} | right]

Max

Min

Position

String

Style: [{pushbutton} | radiobutton | checkbox | edit | text | slider | frame | popupmenu]

Units: [inches | centimeters | normalized | points | {pixels}]

Value

ButtonDownFcn

Clipping: [{on} | off]

Interruptible: [{no} | yes]

Parent

UserData

Visible: [{on} | off]

在用户控制框的各项对象属性中，Style 和 Callhack 是最基本的、最重要的。

·Style（类型）

此属性用一个字符串来指定控制框的类型，属性值共有 8 个：[{pushhutton} | radiobutton | checkbox | edit | text | slider | frame | popupmenu]，其含义依次为按钮、无线按钮、检查框、编辑框、文本框、滑标、边框、弹出式菜单，缺省类型为按钮。各种类型的用户控制框如图 3.6 所示。

按钮可以简化为 Push，它应用得最为广泛，按一下按钮执行某一种特定的操作。弹出式菜单可以简化为 Popup，它需要事先规定几种操作，然后使用鼠标来选择。编辑框提供了使用键盘来输入字符串的功能。文本框的功能是进行说明。滑标的外形和作用类似于一个电位器，它可以完成数据的微调。边框的作用比较简单，基本上是用来给控制框划定一个区域。检查框和无线按钮的作用比较接近，都是用来在两个状态下进行切换，或在一组选项中选择一项。

5.x 版中，还有另外两种控制框的类型：togglebutton 和 listbox，意为切换按钮和列表。

· Callback（回应）

回应的属性值为一个字符串，它代表着执行某种特定的操作，这与用户菜单的回应是完全一样的。一般控制框的回应多为调用某个 M 函数来完成某种操作。

·BackgroundColor（背景色）

背景色用来指定某个控制框的背景颜色，它可以简化为 Back。

·ForegroundColor（前景色）

前景色用来指定某个控制框的前景颜色，它可以简化为 Fore。

· HorizontalAlignment（水平位置）

水平位置规定了沿水平方向上各种控制框采用的对齐方式，其属性值有 3 个 [{left} | center | right]，其含义依次为左对齐、中间对齐、右对齐，缺省值是中间对齐。

属性名可以简化为 Horiz。

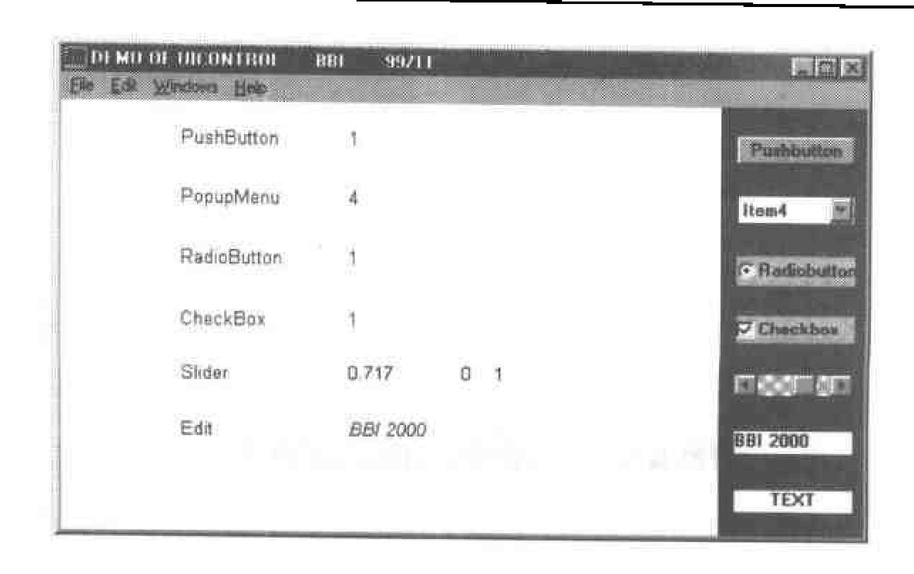


图 3.6 各种类型的用户控制框

· Position（位置）

位置的属性值为向量［left bottom width height］，单位由 Units 属性指定，和其它的位置属性一样，它是用来表明某个控制框的具体位置。

属性名可以简化为 Pos。

· String（字符串）

字符串属性在按钮、无线按钮、检查框和弹出式菜单中用来添加标志，而在编辑框中则代表着返回的字符串。

· Value（数值）

数值属性用在滑标、无线按钮、检查框等类型的控制框内，来返回特定的数值。

· Max（最大值）

用来明确数值属性的最大数值。

· Min（最小值）

用来明确数值属性的最小数值。

· UserData（用户数据）

用户数据的作用与用户菜单中用户数据的作用是完全一样的，合理地使用这个属性，可以非常灵活地在 M 函数和用户控制框之间进行数据的传递。

属性名可以简化为 User。

3.6.2 按钮式界面的设计

按钮式界面（控制框界面）的设计方法是很灵活的，但是一般都采用函数递归调用的方

式，即使用一个M函数来调用自己。掌握按钮式界面设计方法的最好方式是对现成的程序进行模仿，然后根据自己的具体需求来进行设计。典型的按钮式界面的M函数为graf2d.m，读者可以在MATLAB的demos子目录中找到它。

典型的按钮式界面的设计方式如下：

1）建立一个函数，其输入参数为一个功能字符串，它代表着每个按钮的回应。

2）定义缺省的功能字符串为“初始化”。

3）在函数中，首先对功能字符串进行判断，若它是“初始化”，则完成对整个按钮式界面的建立工作；若它代表着某个具体的按钮，则完成该按钮所对应的功能。

4）不同按钮的回应方式是不一样的，具体的方法见下面的实例。

下面是按钮式（控制框式）界面的设计流程图：

```
function [ ] =uictl (action);
%
if nargin<1; action='initialized'; end;
if strcmp (action,'initialized');
```

建立按钮式界面

```
elseif strcmp (action,'button1');
```

执行按钮 1 的功能

```
elseif strcmp (action,'button2');
```

执行按钮 2 的功能

```
elseif strcmp (action,'button3');
```

执行按钮 3 的功能

```
end;
```

3.6.3 实例

在此处介绍一个建立按钮式界面的M函数uidemo.m，它在屏幕上的显示效果见图3.6，读者可通过对它的理解和模仿来建立自己的用户控制框界面，这是一种行之有效的方法。

```
function [ ] =uidemo (action);
%
%   BBI   1999
global H21 H31 H41 H51 H61          %将各句柄定义为全程变量
if nargin<1;                        %缺省值的设定
   action='initialized';
   H21=-1; H31=-1; H41=-1; H51=-1; H61=-1;
```

```
end;
if strcmp (action,'initialized');
  figure ('Name','DEMO OF UICONTROL   BBI   99/11', …
          'Num','off','Units','Normal','Pos', [0.2 0.2 .7 .7], …
          'Color', [1 1 1]);
% 设定坐标轴
axes ('Units','Normal','Pos', [0.08 0.1 .7 .84],'Vis','off');
% 在屏幕右面建立一个暗绿色的区域用来放置各种按钮
H0 = uicontrol ('Style','Frame','Units','Normal', …
               'Pos', [.82 0 .2 1],'Back', [.2 .4 .4]);
% 建立一个按钮
H1 = uicontrol ('Style','Push','Units','Normal', …
               'Position', [.84 .9 .15 .05], …
               'String','Pushbutton', …
               'Callback','uidemo (''Push'');');
% 建立一个弹出式菜单
H2 = uicontrol ('Style','Popup','Units','Normal', …
               'Position', [.84 .8 .15 .05],'Back', [1 1 1], …
               'String', str2mat ('Item1','Item2', …
               'Item3','Item4','Item5'), …
               'Callback','uidemo (''Popup'');');
% 建立一个无线按钮（注：有些 5.2 版的 MATLAB 不支持无线按钮）
H3 = uicontrol ('Style','Radio','Units','Normal',
               'Position', [.84 .7 .15 .05],
               'String','Radiobutton','Back', [.8 .8 .8],
               'Callback','uidemo (''Radio'');');
% 建立一个检查框（注：有些 5.2 版的 MATLAB 不支持检查框）
H4 = uicontrol ('Style','Checkbox','Units','Normal', …
               'Position', [.84 .6 .15 .05], …
               'String','Checkbox','Back', [.8 .8 .8], …
               'Callback','uidemo (''Checkbox'');');
% 建立一个滑标
H5 = uicontrol ('Style','Slider','Units','Normal', …
               'Position', [.84 .5 .15 .04], …
               'Back', [.8 .8 .8],'Value', .618, …
               'Callback','uidemo (''Slider'');');
% 建立一个编辑框
H6 = uicontrol ('Style','Edit','Units','Normal', …
               'Position', [.84 .4 .15 .04], …
```

```
                'Back', [1 1 1],'String','Edit', …
                'Callback','uidemo (''Edit'');');
  % 建立一个文本框
  H7 = uicontrol ('Style','Text','Units','Normal', …
                'Position', [.84 .3 .15 .04], …
                'String','TEXT','Back', [1 1 1]);
  elseif strcmp (action,'Push');        % 按钮的回应
    text (.1, 1,'PushButton','FontSize', 10);
    k = get (gco,'Value'); text (.4, 1, int2str (k),'FontSize', 10);
  elseif strcmp (action,'Popup');       % 弹出式菜单的回应
    text (.1, .88,'PopupMenu','FontSize', 10);
    if H21>0;   delete (H21);   end;
    k = get (gco,'Value');
    H21 = text (.4, .88, int2str (k),'FontSize', 10);
  elseif strcmp (action,'Radio');       %无线按钮的回应
    text (.1, .76,'RadioButton','FontSize', 10);
    if H31>0;   delete (H31);   end;
    k = get (gco,'Value');
    H31 = text (.4, .76, int2str (k),'FontSize', 10);
  elseif strcmp (action,'Checkbox');   % 检查框的回应
    text (.1, .64,'CheckBox','FontSize', 10);
    if H41>0; delete (H41); end;
     k = get (gco,'Value');
    H41 = text (.4, .64, int2str (k),'FontSize', 10);
  elseif strcmp (action,'Slider');      %滑标的回应
    text (.1, .52,'Slider','FontSize', 10);
    if H51>0; delete (H51); end;
    k = get (gco,'Value');
    H51 = text (.4, .52, num2str (k),'FontSize', 10);
    k = get (gco,'Min');   text (.6, .52, int2str (k),'FontSize', 10);
    k = get (gco,'Max');   text (.66, .52, int2str (k),'FontSize', 10);
  elseif strcmp (action,'Edit');        %编辑框的回应
    text (.1, .4,'Edit','FontSize', 10);
    if H61>0; delete (H61); end;   s = get (gco,'String');
    H61 = text (.4, .4, s,'FontSize', 10,'FontAngle','Italic');
  end;
```

将上面的程序输入然后运行该函数，通过选择各按钮，并根据屏幕上的具体显示内容，就可以理解各个按钮的回应方式了。

按钮回应就是执行一个特定的程序段，或调用某个 M 函数；弹出式菜单的回应是一个

数，它与菜单的项数相关，因此根据这个数，就可以确定使用者选择的是哪一项；无线按钮的回应为“真”和“假”，1 代表选中，而 0 代表未选；检查框的情况与无线按钮类似；滑标的回应为一个归一值，而具体的数值可以由这个归一值与最大值和最小值共同来确定；编辑框的回应就是一个字符串。

3.6.4 其它的控制框

在 MATLAB 中还提供了一些其它的控制框，如：uigetfile、uisetfile、uisetfont、uisetcolor 等等。

· uigetfile 函数

uigetfile 函数的功能就是建立一个标准的对话框来打开文件，其用法是：

》[FileName, Path] =uigetfile (Ftype, Name)；Ftype 为字符串，它用来指定文件的类型，通常使用文件的后缀；Name 为对话框的名称。FileName 为被打开的文件名称；Path 为该文件的路径名。

》[F, P] =uigetfile ('*.wav','Open')；打开 WAV 文件，见图 3.7。

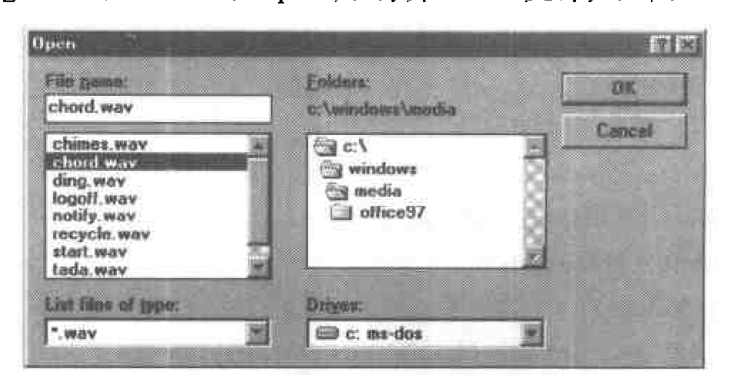


图 3.7　打开文件的对话框（F='chord.wav', P='c:\windows\media\'）

· uiputfile 函数

uiputfile 函数的功能就是建立一个标准的对话框来储存文件，其用法是：

》[FileName, Path] =uiputfile (Ftype, Name)；Ftype 为字符串，它用来指定文件的类型，通常使用文件的后缀；Name 为对话框的名称。FileName 为被储存的文件名称；Path 为该文件的路径名。

》[F, P] =uiputfile ('*.wav','Save as')；　储存 WAV 格式的文件。

· uisetfont 函数

uisetfont 函数的功能就是建立一个标准的对话框来改变字体，其基本用法是：

》H=uisetfont (H1)；　H1 为要改变的字体所对应的句柄，即坐标对象的句柄或文本对象的句柄，而 H 为返回的句柄。

》H1 = text（.1，.5，'abc'）；H2 = text（.1，.7，'efg'）；H = uisetfont（H1）；
仅改变'abc'的字体。

· uisetcolor 函数

uisetcolor 函数的功能就是建立一个标准的对话框来改变颜色，其基本用法是：

》H = uisetcolor（H1）；　　H1 为要改变的颜色所对应的句柄，如图形窗口对象、坐标对象、文本对象、线条对象的句柄，而 H 为返回的句柄。

参考文献

[1]（美）Duane Hanselman，Bruce Littlefield．精通 MATLAB．西安：西安交通大学出版社，1988

[2] 张培强．MATLAB 语言．合肥：中国科技大学出版社，1995

[3] 薛定宇．控制系统计算机辅助设计．北京：清华大学出版社，1996

第四章　电子系统仿真的常用工具箱

在本章内对电子系统仿真过程中经常使用的工具箱进行简明扼要地介绍，其中包括信号工具箱、符号运算工具箱、系统仿真工具箱、DSP 工具箱、通信工具箱、图象工具箱和编译工具箱。

4.1　SIGNAL 工具箱

SIGNAL 工具箱的全称为信号处理工具箱，它是进行电子系统仿真必不可少的工具箱，存在着许多不同的版本，与 4.2 版 MATLAB 配套的是 3.0b 版，与 5.2 版 MATLAB 配套的是 4.1 版，与 5.3 版 MATLAB 配套的是 4.2 版。在本节内介绍 3.0b 版的主要内容。

4.1.1　波形发生器

· diric 函数

diric 函数用于建立狄里赫莱函数，该函数的定义是 $d(t)=\frac{\sin(N\cdot t/2)}{N\sin(t/2)}$，其中 N 表示在 $[t_0\ t_0+2\pi]$ 区间内，d（t）取极大值的数目。diric 函数的用法是：

》d = diric（t，N）；　　　t 为向量或矩阵，N 为函数的参数，d 为函数值。

· sawtooth 函数

sawtooth 函数用于建立锯齿波函数，其用法是：

》x = sawtooth（t）；　　　t 为向量或矩阵，此函数的周期为 2π，在每个周期内函数的图形为一条斜线，函数值在［-1 1］范围内。

》x = sawtooth（t，w）；　　　t 为向量或矩阵，w 的取值范围是［0 1］，此函数的周期为 2π，在［0 w * π］区间内函数的图形为一条上升的直线，函数值从 -1 增加至 +1，而在［w * π π］区间内函数的图形为一条下降的直线，函数值从 +1 减小至 -1。

· sinc 函数

sinc 函数的定义 $\text{sinc}(t)=\begin{cases}\sin(\pi\cdot t)/(\pi\cdot t), & t\neq 0\\ 1, & t=0\end{cases}$，其用法是：

》x = sinc（t）；　　　t 为向量或矩阵，x 为函数值。

· square 函数

square 函数用于建立方波函数，其用法是：

》x = square (t)；　　t 为时间向量或矩阵，函数的周期为 2π，在前半周期内函数值为 +1，而在后半周期内函数值为 -1。

》x = square (t, p)；　　t 为时间向量或矩阵，函数的周期为 2π，在 p% 的周期内函数值为 +1，而在剩下的周期内函数值为 -1。

4.1.2 IIR 滤波器（无限冲击响应滤波器）设计

在 MATLAB 中，IIR 滤波器的设计工作就是要确定滤波器的两个系数向量 a 和 b。IIR 数字滤波器的转移函数为：

$$H(z) = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2} + \cdots + b_n z^{-n+1}}{1 + a_2 z^{-1} + a_3 z^{-2} + \cdots + a_m z^{-m+1}}$$

其中，分子的系数向量记为 b，分母的系数向量记为 a。数字滤波的实现是采用差分方程的方式：

$$y_n = b_1 x_n + b_2 x_{n-1} + b_3 x_{n-2} + \cdots + b_m x_{n-m+1}$$
$$- a_2 y_{n-1} - a_3 y_{n-2} - \cdots - a_k y_{n-k+1}$$

由于 IIR 数字滤波器的设计是借助于模拟滤波器的设计方法来完成的，也就是说设计 IIR 数字滤波器的 M 函数亦可以用来设计模拟滤波器。模拟滤波器的转移函数为：

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_1 s^n + b_2 s^{n-1} + b_3 s^{n-2} + \cdots + b_{n-1} s + b_n}{a_1 s^m + a_2 s^{m-1} + a_3 s^{m-2} + \cdots + a_{m-1} s + a_m}$$

其中，分子的系数向量记为 b，分母的系数向量记为 a。模拟滤波器就是用电阻、电容和电感元件来实现滤波器的转移函数。

·butter 函数

butter 函数用于设计巴特伍兹数字滤波器和巴特伍兹模拟滤波器，巴特伍兹滤波器又称为最大平坦滤波器，巴特伍兹低通滤波器的幅频特性如图 4.1 所示，图中的横坐标为归一化的角频率。

butter 函数的用法是：

》[b, a] = butter (N, Wn)；　　输出向量 a 和 b 为 N 阶巴特伍兹数字低通滤波器的系数。输入参数 Wn 是截止角频率的归一值，对应巴特伍兹滤波器来说截止频率就是 3dB 点的频率，$W_n = 1$ 对应于采样角频率的一半，故 Wn 的取值范围是 [0 1]。

》[b, a] = butter (N, Wn, 'high')；　　输出向量 a 和 b 为 N 阶巴特伍兹数字高通滤波 器的系数。输入参数 Wn 是截止角频率的归一值，Wn 的取值范围是 [0 1]。

》[b, a] = butter (N, [W1 W2])；　　输出向量 a 和 b 为偶数阶巴特伍兹数字带通滤 波器的系数。输入参数 W1 和 W2 分别为通带下限和上限截止角频率的归一值，0 < W1 < W2 < 1。

》[b, a] = butter (N, [W1 W2], 'stop')；　　输出向量 a 和 b 为偶数阶巴特伍兹数字

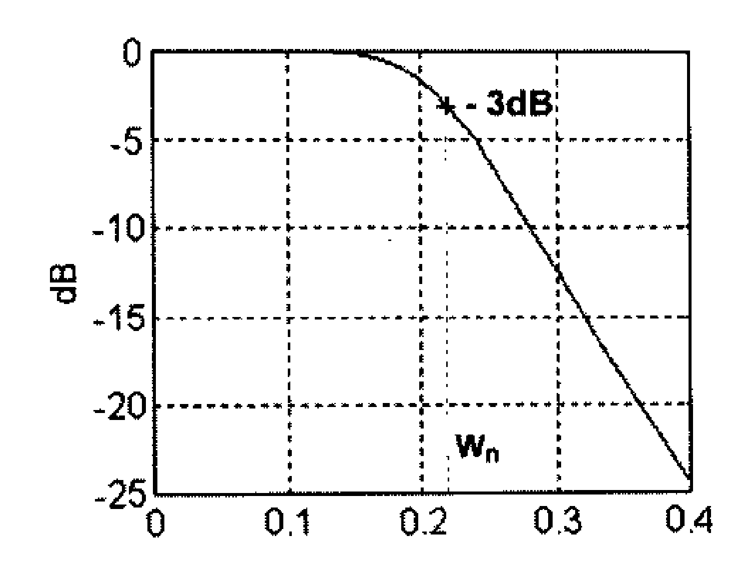


图 4.1 巴特伍兹低通滤波器的幅频特性

带阻滤波器的系数。输入参数 W1 和 W2 为通带下限和上限截止角频率的归一值，0 < W1 < W2 < 1。

〉[b, a] =butter (N, Wn,'s')；　　输出向量 a 和 b 为 N 阶巴特伍兹模拟低通滤波器的系数，对于模拟滤波器来说 Wn 是截止角频率，但不是归一值，即 Wn 可以大于 1。

〉[b, a] =butter (N, Wn,'high','s')；　　输出向量 a 和 b 为 N 阶巴特伍兹模拟高通滤波器的系数，对于模拟滤波器来说 Wn 是截止角频率，非归一值，即 Wn 可以大于 1。

〉[b, a] =butter (N, [W1 W2],'s')；　　输出向量 a 和 b 为 N 阶巴特伍兹模拟带通 滤波器的系数，对于模拟滤波器来说 W1 和 W2 是通带的下限和上限截止角频率，非归一值。

〉[b, a] =butter (N, [W1 W2],'stop','s')；　　输出向量 a 和 b 为 N 阶巴特伍兹模拟带阻滤波器的系数，对于模拟滤波器来说 W1 和 W2 是通带的下限和上限截止角频率，非归一值。

·buttord 函数

buttord 函数根据通带的衰减和阻带的衰减来确定巴特伍兹滤波器的阶数，其用法是：

〉[N, Wn] =buttord (Wp, Ws, Rp, Rs)；

N 为巴特伍兹数字滤波器的阶数，Wn 为 3dB 点的归一角频率。Ws 和 Wp 分别是通带截止角频率和阻带截止角频率，见图 4.2，它们都是归一角频率，并且有 0<Wp<Ws<1。

Rp 是通带的衰减，而 Rs 为阻带的衰减，衰减量的单位为 dB。

》[N, Wn] =buttord (Wp, Ws, Rp, Rs,'s');

N 为巴特伍兹模拟滤波器的阶数，Wn 为 3dB 点的归一角频率。Ws 和 Wp 分别是通带和阻带的截止角频率（非归一值），Rp 和 Rs 分别为通带和阻带的衰减，单位为 dB。

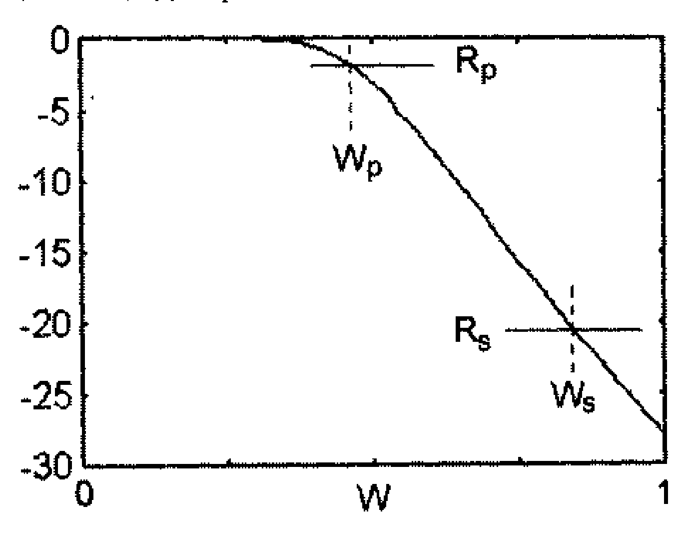


图 4.2　滤波器的设计

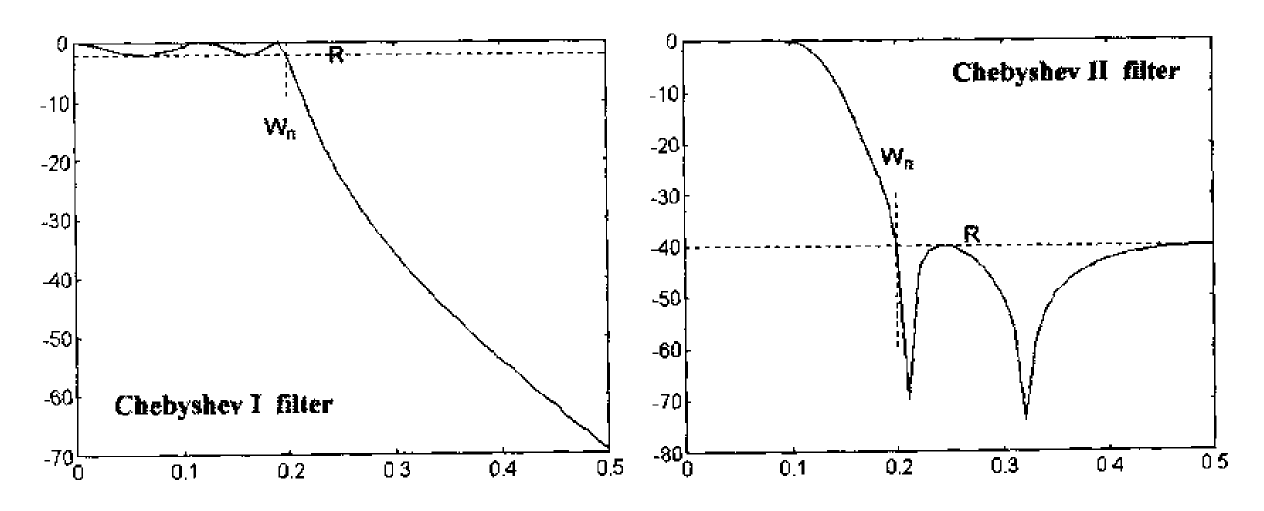


图 4.3　切比雪夫 I 型和 II 型滤波器的幅频特性

· cheby1 函数

cheby1 函数用于设计切比雪夫 I 型数字滤波器和模拟滤波器，此型滤波器的特征是在其通带内，幅频特性在一定的范围内起伏变化，称为波纹，见图 4.3。cheby1 函数的用法与巴特伍兹滤波器的设计类似，但增加一个输入参数 R 来描述波纹，单位 dB。

》[b, a] =cheby1 (N, Wn);　　N 阶切比雪夫 I 型数字低通滤波器

》[b, a] =cheby1 (N, Wn,'high');　　N 阶切比雪夫 I 型数字高通滤波器

》[h, a] =cheby1 (N, [W1 W2]);　　N 阶切比雪夫 I 型数字带通滤波器

》[b, a] =cheby1 (N, [W1 W2],'stop');　　N 阶切比雪夫 I 型数字带阻滤波器

》[b, a] =cheby1 (N, Wn,'s');　　N 阶切比雪夫 I 型模拟低通滤波器

》[b, a] =cheby1 (N, Wn,'high','s');　　N 阶切比雪夫 I 型模拟高通滤波器
》[b, a] =cheby1 (N, [W1 W2],'s');　　N 阶切比雪夫 I 型模拟带通滤波器
》[b,a]=cheby1(N,[W1 W2],'stop','s');　　N 阶切比雪夫 I 型模拟带阻滤波器

· cheb1ord 函数

cheb1ord 函数根据通带的衰减和阻带的衰减来确定切比雪夫 I 型滤波器的阶数，其用法与 buttord 函数类似：

》[N,Wn]=cheb1ord(Wp,Ws,Rp,Rs);　　N 为切比雪夫 I 型数字滤波器的阶数，W_n 为转折角频率（归一值）。

》[N,Wn]=cheb1ord(Wp,Ws,Rp,Rs,'s');　N 为切比雪夫 I 型模拟滤波器的阶数，W_n 为转折角频率（非归一值）。

· cheby2 函数

cheby2 函数用于设计切比雪夫 II 型数字滤波器和模拟滤波器，此型滤波器的特征是在其通带内幅频特性是平坦的，而在其阻带内幅频特性成起伏变化（波纹），见图 4.2。cheby2 函数的用法与 cheby1 函数的用法类似。

》[b, a] =cheby2 (N, Wn);　　N 阶切比雪夫 II 型数字低通滤波器
》[b, a] =cheby2 (N, Wn,'high');　　N 阶切比雪夫 II 型数字高通滤波器
》[b, a] =cheby2 (N, [W1 W2]);　　N 阶切比雪夫 II 型数字带通滤波器
》[b, a] =cheby2 (N, [W1 W2],'stop');N 阶切比雪夫 II 型数字带阻滤波器
》[b, a] =cheby2 (N, Wn,'s');　　N 阶切比雪夫 II 型模拟低通滤波器
》[b, a] =cheby2 (N, Wn,'high','s');　N 阶切比雪夫 II 型模拟高通滤波器
》[b, a] =cheby2 (N, [W1 W2],'s');　N 阶切比雪夫 II 型模拟带通滤波器
》[b,a]=cheby2(N,[W1 W2],'stop','s');　N 阶切比雪夫 II 型模拟带阻滤波器

· cheb2ord 函数

cheb2ord 函数根据通带的衰减和阻带的衰减来确定切比雪夫 II 型滤波器的阶数，其用法与 cheb1ord 函数的用法相同：

》[N,Wn]=cheb2ord(Wp,Ws,Rp,Rs);　　N 为切比雪夫 II 型数字滤波器的阶数，W_n 为转折角频率（归一值）。

》[N,Wn]=cheb2ord(Wp,Ws,Rp,Rs,'s');　N 为切比雪夫 II 型模拟滤波器的阶数，W_n 为转折角频率（非归一值）。

· ellip 函数

ellip 函数用于设计椭圆数字滤波器和模拟滤波器，这种滤波器的特征是在其通带内幅频特性成起伏变化，这种变化用波纹 Rp 来描述，而在其阻带内幅频特性也成起伏变化，这种变化用波纹 Rs 来描述，波纹的单位为 dB。ellip 函数的基本用法是：

》[b, a] =ellip (N, Rp, Rs, Wn);　　N 阶椭圆数字低通滤波器

〉[b, a] =ellip (N, Rp, Rs, Wn,'high'); N 阶椭圆数字高通滤波器
〉[b, a] =ellip (N, Rp, Rs, [W1 W2]); N 阶椭圆数字带通滤波器
〉[b, a] =ellip (N, Rp, Rs, [W1 W2],'stop'); N 阶椭圆数字带阻滤波器
〉[b, a] =ellip (N, Rp, Rs, Wn,'s'); N 阶椭圆模拟低通滤波器
〉[b, a] =ellip (N, Rp, Rs, Wn,'high','s'); N 阶椭圆模拟高通滤波器
〉[b, a] =ellip (N, Rp, Rs, [W1 W2],'s'); N 阶椭圆模拟带通滤波器
〉[b, a] =ellip (N, Rp, Rs, [W1 W2],'stop','s'); N 阶椭圆模拟带阻滤波器

· ellipord 函数

ellipord函数根据通带的衰减和阻带的衰减来确定椭圆滤波器的阶数，其用法与cheb1ord 函数的用法相同：

〉[N, Wn] =ellip (Wp, Ws, Rp, Rs); N 为椭圆数字滤波器的阶数，W_n 为 转折角频率（归一值）。

〉[N, Wn] =ellip(Wp, Ws, Rp, Rs,'s'); N 为椭圆模拟滤波器的阶数，W_n 为 转折角频率（非归一值）。

4.1.3 FIR 滤波器（有限冲击响应滤波器）设计

FIR 滤波器的主要特点是其相频特性是线性的，故广泛地应用在数字信号的传输系统内。在 MATLAB 中，FIR 滤波器的设计工作就是来确定滤波器的系数向量 b。FIR 数字滤波器的转移函数为：

$$H(z) = b_1 + b_2 z^{-1} + b_3 z^{-2} + \cdots + b_n z^{-n+1}$$

数字滤波的实现也采用差分方程的方式：$y_n = b_1 x_n + b_2 x_{n-1} + b_3 x_{n-2} + \cdots + b_m x_{n-m+1}$

· fir1 函数

fir1 函数使用窗函数法来设计 FIR 数字滤波器，其输入参数的含义与 IIR 滤波器相同，fir1 函数基本用法是：

〉b=fir1 (N, Wn); N 阶低通 FIR 滤波器（使用汉明窗）
〉b=fir1 (N, Wn,'high'); N 阶高通 FIR 滤波器（使用汉明窗）
〉b=fir1 (N, [W1 W2]); N 阶带通 FIR 滤波器（使用汉明窗）
〉b=fir1 (N, [W1 W2],'stop'); N 阶带阻 FIR 滤波器（使用汉明窗）

若要使用其它的窗函数（见表 4.1），则可用输入参数来说明，例如：

〉b=fir1 (N, Wn, chebwin (N+1, R));
N 阶低通 FIR 滤波器（使用切比雪夫窗）
〉b=fir1 (N, Wn,'high', bartlett (N+1));
N 阶高通 FIR 滤波器（使用巴特莱特窗）

·fir2 函数

fir2 函数使用窗函数法来设计任意幅频特性的 FIR 数字滤波器，其用法是：

〉h=fir2 (N, F, M); 使用汉明窗设计 N 阶 FIR 滤波器，滤波器的幅频特性

由向量 F 和 M 来描述，F 为归一化频率，其元素的数值范围是 [0 1]，M 则为幅度值。

若要使用其它的窗函数（见表 4.1），则可用输入参数来说明，例如：

〉b=fir2 (N, F, M, chebwin (N+1, R));　使用切比雪夫窗设计 N 阶 FIR 滤波器

〉b=fir2 (N, F, M, bartlett (N+1));　使用巴特莱特窗设计 N 阶 FIR 滤波器

表 4.1　信号工具箱中的各种窗函数

窗函数	说　明
bartlett (n)	n 个数据的巴特莱特窗
blackman (n)	n 个数据的布莱克曼窗
boxcar (n)	n 个数据的矩形窗
chebwin (n, R)	n 个数据的切比雪夫窗，波纹为 R 分贝，n 为奇数
hamming (n)	n 个数据的汉明窗
hanning (n)	n 个数据的汉宁窗
kaiser (n, beta)	n 个数据 beta 参数的凯泽窗
triang (n)	n 个数据的三角窗

· firls 函数

firls 函数使用最小二乘方逼近法来设计任意幅频特性的 FIR 数字滤波器，其主要用法是：

〉b=firls (N, F, M);　使用最小二乘方逼近法来设计 N 阶 FIR 滤波器，滤波器的幅频特性由向量 F 和 M 来描述，F 为归一化频率，其元素的数值范围是 [0 1]，M 则为幅度值。

〉b=firls (N, F, M, W);　使用权向量 W 来估计误差。

〉b=firls (N, F, M,'hilbert');　使用最小二乘方逼近法来设计 N 阶希尔伯特变换器。

〉b=firls (N, F, M,'differentiator');　使用最小二乘方逼近法来设计 N 阶微分器。

· remez 函数

remez 函数使用切比雪夫逼近法（具体称为 Parks－McClellan 算法）来设计任意幅频特性的 FIR 数字滤波器，其主要用法是：

〉b=remez (N, F, M);　使用切比雪夫逼近法来设计 N 阶 FIR 滤波器，滤波器的幅频特性由向量 F 和 M 来描述，F 为归一化频率，其元素的数值范围是 [0 1]，M 则为幅度值。

〉b=remez (N, F, M, W);　使用权向量 W 来估计误差。

〉b=remez (N, F, M,'hilbert');　使用切比雪夫逼近法来设计 N 阶希尔伯特变换器。

〉b=remez (N, F, M,'differentiator');　使用切比雪夫逼近法来设计 N 阶微分器。

· remezord 函数

remezord 函数与 remez 函数联合使用，用来确定 remez 函数的各项输入参数，其主要用法是：

》[N, Fo, Mo, W] =remezord (F, M, DEV, Fs);

滤波器的幅频特性由向量 F 和 M 来描述，注意此处 F 为频率而不是归一化频率，M 则为幅度值；DEV 为允许的最大离差（波纹），F_s 为采样频率。各输出参数则作为 remez 函数的输入参数。

4.1.4　滤波器的分析与实现

这部分介绍的是与滤波器使用有关的 M 函数。在 MATLAB 中，数字滤波的实现通常是使用 filter 函数，关于该函数见 2.8.5

· freqz 函数

freqz 函数使用 Z 变换来分析数字滤波器的频率响应，当一个数字滤波器设计完毕之后可以使用此函数来确定该滤波器是否达到了预期的效果，其主要用法是：

》freqz (b, a);　　输入参数为数字滤波器的系数向量。函数执行后在屏幕上同时显示出该滤波器的幅频特性曲线和相频特性曲线，注意：图中横坐标的单位为归一化的角频率，它等于 1 时，实际频率为采样频率的一半。

》h=freqz (b, a, f, fs);　　输入参数 b 和 a 为数字滤波器的系数向量，f 为频率向量，单位为 Hz，f_s 为采样频率，单位为 Hz。输出参数 h 就是该数字滤波器的转移函数，得到了转移函数 h 之后便可以绘制出各种形式的幅频特性曲线和相频特性曲线，如对数坐标、分贝坐标等等。

· freqs 函数

freqs 函数根据模拟网络的转移函数使用拉普拉斯变换来分析其频率响应，模拟网络的转移函数可以由系数向量 b 和 a 来描述，其主要用法是：

》freqs (h, a);　　输入参数 b 和 a 为模拟网络转移函数的系数向量。函数执行后在屏幕上同时显示出该网络的幅频特性曲线和相频特性曲线，注意：图中横坐标的单位为角频率，单位为弧度。

》g=freqs (b, a, w);　　输入参数 b 和 a 为模拟网络转移函数的系数向量，w 为角频率向量。输出参数 g 就是该网络的转移函数，得到了转移函数 g 之后便可以绘制出各种形式的幅频特性曲线和相频特性曲线，如对数坐标、分贝坐标等等。

· impz 函数

impz 函数给出数字滤波器的冲击响应，其主要用法是：

〉impz（b，a）；　　输入参数为数字滤波器的系数向量。函数执行后在屏幕上显示 出该滤波器的冲击响应。

〉[h，t] =impz（b，a）；　　输入参数为数字滤波器的系数向量。输出参数 h 为该滤波器的冲击响应序列，而 t 为时间序列。

· grpdelay 函数

grpdelay 函数给出数字滤波器的群时延，其主要用法是：

〉grpdelay（b，a）；　　输入参数为数字滤波器的系数向量。函数执行后在屏幕上显示出该滤波器的群时延。

〉[gd，W] =grpdelay（b，a）；　　输入参数为数字滤波器的系数向量。输出参数 gd 为该滤波器的群时延，而 W 为归一角频率，单位为弧度。

· zplane 函数

zplane 函数用来绘制 Z 平面内的零极点图，其用法是：

〉zplane（Z，P）；　　Z 为列向量，它表示零点的位置；P 为列向量，它表示极点的位置。函数执行后在屏幕上显示出相应的零极点图；在图中 x 表示极点，o 表示零点。

〉zplane（b，a）；　　输入参数为转移函数的系数行向量，函数在执行的过程中先计算出零点和极点，然后再显示出零极点图。

·filtfilt 函数

filtfilt 函数将信号序列滤波两次，一次为正向滤波，另一次为反向滤波，这样做的结果是实现了零相移，同时将滤波器的阶数等效地提高了一倍。它的用法是：

〉y=filtfilt（b，a，x）；　　x 是信号序列，y 是滤波器的输出序列，b 和 a 是数字 滤波器的系数向量。

4.1.5 线性系统变换

信号工具箱内设置了很多在线性系统之间进行系统变换的函数，见表 4.2。

4.1.6 变换

· czt 函数

czt 函数用来在指定的路径上对信号序列进行 Z 变换，它的用法是：

〉g=czt（x，m，w，a）；　　x 是信号序列，m、w 和 a 均为标量，用来确定 z 平面上的路径：z=a * w.^（-（0：m-1））。输出序列为 g。

〉g=czt（x）；　　x 是信号序列，输出序列为 g，采用缺省的路径，其结果与 fft（x）相同。

·离散余弦变换（DCT）

表 4.2 信号工具箱中的各种线性系统变换函数

变换函数	说明
K = poly2rc(A)	将格型滤波器的多项式系数向量 A 转换为反射系数向量 K
A = rc2poly(K)	将格型滤波器的反射系数向量 K 转换为多项式系数向量 A
[A, B, C, D] = sos2ss(sos)	将分段二阶模型参数 sos 换算转换为状态变量矩阵 A, B, C, D
[Z, P, K] = sos2zp(sos)	将分段二阶模型参数 sos 换算转换为复频域内的零极点方程
sos = ss2sos(A, B, C, D)	将状态变量矩阵 A, B, C, D 换算转换为分段二阶模型参数 sos
[b, a] = ss2tf(A, B, C, D, iu)	将状态变量矩阵转换为复频域内的转移函数
[Z, P, K] = ss2zp(A, B, C, D, iu)	将状态变量矩阵转换为复频域内的零极点方程
[A, B, C, D] = tf2ss(b, a)	将复频域内的转移函数转换为状态变量矩阵
[Z, P, K] = tf2zp(b, a)	将复频域内的转移函数转换为零极点方程
[b, a] = zp2tf(Z, P, K)	将复频域内的零极点方程转换为转移函数
[A, B, C, D] = zp2ss(Z, P, K)	将复频域内的零极点方程转换为状态变量矩阵
sos = zp2sos(Z, P, K)	将复频域内的零极点方程换算转换为分段二阶模型参数 sos

〉y = dct (x);　　y 是 x 的离散余弦变换。

〉x = idct (y);　　x 是 y 的离散余弦逆变换。

· 离散希尔伯特变换

〉y = hilbert (x);　　x 是实数向量或矩阵。y 的虚部为离散希尔伯特变换的结果，实部仍为输入的实数 x，y 称为 x 的解析信号。

4.1.7 信号统计处理

在信号处理工具箱内提供了若干分析平稳随机信号的 M 函数。

· psd 函数

psd 函数使用 Welch 的平均周期图法（又称为加权交叠平均法）来估计一个随机信号序列的自功率谱，它的基本用法是：

〉Pxx = psd (x, nfft, fs, window);　　x 输入的信号序列，Pxx 是它的自功率谱。nfft 是 FFT 的长度，fs 为采样频率，window 为使用的窗函数。

〉Pxx = psd (x);　　使用缺省的参数来进行估计信号的功率谱。当序列的长度小于 256 时，nfft = length (x)，否则 nfft = 256；fs = 2（归一值）；使用汉明窗。

〉psd (x, y, nfft, fs, window);　　绘制功率谱曲线。

· csd 函数

csd 函数使用 Welch 的平均周期图法（又称为加权交叠平均法）来估计两个随机序列间的交叉功率谱，它的基本用法是：

》Pxy=csd (x, y, nfft, fs, window);

x和y是输入的两个序列，Pxy是两个随机信号序列间的交叉功率谱。nfft是FFT的长度，fs为采样频率，window为使用的窗函数。

》Pxy=csd (x, y);

使用缺省的参数来进行估计交叉功率谱。当序列的长度小于256时，nfft=length (x)，否则nfft=256；fs=2 (归一值)；使用汉明窗。

》csd (x, y, nfft, fs, window);　绘制交叉功率谱曲线。

· cohere函数

cohere函数使用Welch的平均周期图法（又称为加权交叠平均法）来估计两个随机序列之间的相干函数，它的基本用法是：

》Cxy=cohere (x, y, nfft, fs, window);

x和y是输入的两个序列，Cxy是相干函数，Cxy=（ahs (Pxy) .^2) ./ (Pxx. * Pyy)，其数值范围在0和1之间。nfft是FFT的长度，fs为采样频率，window为使用的窗函数。

》Cxy=cohere (x, y);

使用缺省的参数来进行估计相干函数。当序列的长度小于256时，nfft=length (x)，否则nfft=256；f_s=2 (归一值)；使用汉明窗。

·cohere (x, y, nfft, fs, window);　绘制相干函数曲线。

· tfe函数

tfe函数使用Welch的平均周期图法（又称为加权交叠平均法）来估计输入随机信号序列和输出信号序列之间的转移函数，它的基本用法是：

》Txy=tfe (x, y, nfft, fs, window);

x是输入的随机信号序列，y是输出的信号序列，Txy是对转移函数的估计，Txy=Pxy./Pxx 。nfft是FFT的长度，f_s为采样频率，window为使用的窗函数。

》Txy=tfe (x, y);

使用缺省的参数来进行估计转移函数。当序列的长度小于256时，nfft=length (x)，否则nfft=256；f_s=2 (归一值)；使用汉明窗。

》tfe (x, y, nfft, fs, window);　绘制转移函数曲线。

· xcorr函数

xcorr函数用来估计两个随机序列的互相关函数或一个随机序列的自相关函数，它的基本用法是：

》rxy=xcorr (x, y);　　x和y是两个随机信号序列，rxy是对互相关函数的估计。

》rxx=xcorr (x);　　x是一个随机信号序列，rxx是对自相关函数的估计。

·xcov函数

xcov 函数用来估计两个随机序列的互协方差函数或一个随机序列的自协方差函数，它的基本用法是：

》cxy = xcov (x, y);　　x 和 y 是两个随机信号序列，cxy 是对互协方差函数的估计。

》cxx = xcov (x);　　x 是一个随机信号序列，cxx 是对自协方差函数的估计。

4.1.8 特殊操作

· decimate 函数

decimate 函数的功能是使用低通滤波器对离散数据进行较低速率的重新采样，这个过程称为抽选，decimate 函数的用法是：

》y = decimate (x, R, N);　　x 是离散序列，y 是抽选后的离散序列，R 是输入的采样率与输出的采样率之比，R>1。数据抽选使用的是 N 阶切比雪夫低通滤波器。

》y = decimate (x, R);　　使用 8 阶的切比雪夫 I 型低通滤波器进行数据的抽选。

》y = decimate (x, R, N,'fir');　使用 N 阶的 FIR 低通滤波器进行数据的抽选。

》y = decimate (x, R,'fir');　　使用 30 阶的 FIR 低通滤波器进行数据的抽选。

· interp 函数

interp 函数的功能是使用低通插值的方法对离散数据进行较高速率的重新采样，其基本用法是：

》y = decimate (x, R);　　x 是输入的离散序列，y 是输出的离散序列，R 为输出的采样率与输入的采样率之比，R>1。

· resemple 函数

resemple 函数的功能对离散数据采用不同的采样速率进行重新采样，其中使用了凯泽低通滤波器作为反混迭滤波器，其用法是：

》y = resemple (x, p, q, n, beta);　x 是输入的离散序列，y 是输出的离散序列。p 和 q 均为整数，输出采样率和输入采样率之比为 p/q；n 为一整数，重新采样是针对 xi 两侧各 n 个数据而 进行的；beta 为凯泽滤波器的参数。

》y = resemple (x, p, q, n);　　使用缺省值：beta = 5 。

》y = resemple (x, p, q);　　使用缺省值：beta = 5，n = 10 。

》y = resemple (x, p, q, b);　　使用系数向量 b 描述的滤波器作为反混迭滤波器。

· modulate 函数

modulate 函数的功能是对输入信号进行调制，其用法是：

》y = modulate (x, fc, fs, method, opt);

x 是输入信号，y 是调制后的输出信号，f_c 为载波频率、f_s 为采样频率，频率的单位为 Hz，字符串 method 用来说明调制的方式，字符串 opt 用来说明某种调制方式的功能选项，见表 4.3。

表 4.3 调制方式

method	说　明	opt
am	抑制载波的双边带调幅	-
amdsb - sc	抑制载波的双边带调幅	-
amdsb - tc	调幅（双边带、传输载波）	输入信号的压缩比
amssb	单边带调幅	-
fm	频率调制（调频）	调频指数
pm	相位调制（调相）	调相指数
pwm	脉宽调制	centred, left
ptm	脉冲时间调制	0 至 1 之间的一个数
qam	正交调幅	矩阵的大小

· demod 函数

demod 函数的功能是对输入的调制信号进行解调，其用法是：

〉 x = modulate (y, fc, fs, method, opt);

y 是输入的调制信号，x 是解调后的信号，fc 为载波频率，fs 为采样频率，频率的单位为 Hz，字符串 method 用来说明调制的方式，字符串 opt 用来说明某种调制方式的功能选项，见表 4.3。

· vco 函数

vco 函数的功能是一个模拟的压控振荡器，其用法是：

〉 y = vco (x, fc, fs);

y 是压控振荡器的输出信号，x 是控制电压向量，fc 为压控振荡器的固有振荡频率，fs 为采样频率，频率的单位均为 Hz。控制电压的数值范围是 [-1　1]，当控制电压为 0 时，振荡器的振荡频率就是其固有频率；当控制电压为 1 时，振荡频率为固有频率的 2 倍；当控制电压为 -1 时，振荡频率为固有频率的 1/2。输出的 y 向量的大小与输入的 x 向量相同，它为正弦波，按 fs 的数值进行采样。

〉 y = vco (x, [fmin fmax], fs);　　振荡频率变化范围是 [fmin fmax]。

4.1.9 其它

在信号工具箱内还有一些其它用途的 M 函数，见表 4.4。

4.1.10 演示

信号工具箱内设有 4 个专门用于演示的 M 函数，观看演示是掌握信号工具箱的一种好方法。

ctzdemo　-　FFT 和 CZT 的演示

filtdemo　-　滤波器设计的演示

moddemo　-　调制与解调的演示

sosdemo　-　分段二阶模型的演示

表 4.4　信号工具箱中的一些 M 函数

函　数	说　明
[b, a] = besself (N, Wn, s)	设计模拟贝塞尔滤波器
[z, p, k] = besselap (N)	设计模拟的贝塞尔原型滤波器
[z, p, k] = buttap (N)	设计模拟的巴特伍兹原型滤波器
[z, p, k] = cheb1ap (N)	设计模拟的切比雪夫 I 型原型滤波器
[z, p, k] = cheb2ap (N)	设计模拟的切比雪夫 II 型原型滤波器
[z, p, k] = ellipap (N)	设计模拟的椭圆原型滤波器
[b, a] = lp2bp (b0, a0, W0)	将原型滤波器变换为带通滤波器
[b, a] = lp2bs (b0, a0, W0)	将原型滤波器变换为带阻滤波器
[b, a] = lp2hp (b0, a0, W0)	将原型滤波器变换为高通滤波器
[b, a] = lp2lp (b0, a0, W0)	将原型滤波器变换为低通滤波器
xcorr2 (x, y)	二维的交叉相关函数
strips	绘制带状数据图（分段绘图）

4.2　SYMBOLIC 工具箱

SYMBOLIC 工具箱的全称为符号运算工具箱，存在着许多不同的版本，与 4.2 版 MATLAB 配套的是 1.1a 版，与 5.2 版 MATLAB 配套的是 2.0.1 版，与 5.3 版 MATLAB 配 套的是 2.1 版。不同版本之间存在着较大的区别，对于电子系统仿真来说，最好使用 2.0 以上的版本，其原因是在 1.1 版进行四则运算的方式与通常的书写习惯不同。在本节内介绍 2.0.1 版的主要内容，故列举的一些例子在 1.1 版内可能失效。

MATLAB 中的符号运算功能是建立在一种名为 Maple 的软件基础之上的，也就是说符号运算是在该软件内进行的，而将结果返回到 MATLAB 的命令窗口之中。

在电子系统仿真的过程中，符号运算工具箱的主要作用是用来推导复杂的公式，在这点上其它方法是很难与它媲美的，因此符号运算工具箱也是进行电子系统仿真的必备工具箱。

4.2.1　基本操作

符号运算的对象是符号变量，同时也支持一般的标量、向量、矩阵，为了将符号变量与其它变量相区别，在进行符号运算之前，必须对符号变量进行说明。

· sym 函数

sym 函数的作用是对符号变量进行说明，其基本用法是：

》x = sym ('x');	说明 x 为符号变量。
》x = sym ('x','real');	说明 x 为符号变量，且为实数。
》x = sym ('x','positive');	说明 x 为符号变量，且为正实数。
》x = sym ('x','unreal');	说明 x 为符号变量，并可以是复数。
》a = sym ('1/3');	说明 a 为符号数字，其数值为 1/3。

· syms 函数

syms 函数的作用是对多个符号变量进行说明，其用法是：

》syms x y z a b ;　　　　定义了 5 个符号变量。

》syms x y z a b real;　　　　定义了 5 个符号变量，且为实数。

》syms x y z a b positive;　　　　定义了 5 个符号变量，且为正实数。

显然与 sym 函数相比，syms 函数使用起来更为方便。符号变量进行各种运算（四则运算、乘方、开方、对数、函数运算等）后生成的符号表达式和符号变量以及由符号变量组成的符号矩阵都不需要再进行说明了。

》z=sqrt (x * x+y * y)　　　　返回的是 z = (x^2 + y^2) ^ (1/2)

· findsym 函数

findsym 函数的功能是在符号表达式和符号矩阵中寻找符号变量，如：

》z=sqrt (x * x+y * y); findsym (z)　　返回的是 ans=x, y

· pretty 函数

pretty 函数的功能是尽可能按照一般的书写习惯来表示符号表达式，如：

》syms x; pretty (1-0.5 * x * x)　　　返回的是 $1 - 1/2x^2$

· simplify 函数

simplify 函数的功能为对符号表达式进行化简，如：

》syms x; simplify (sin (x) ^2 + cos (x) ^2)　　　返回的是 ans=1

· expand 函数

expand 函数的功能是对符号表达式进行展开，如：

》syms x y; expand (sin (x+y))　　　返回的是 ans= sin(x)*cos(y) + cos(x)*sin(y)

》syms x; y=expand ((x+1) ^3)　　返回的是 y=x^3+3 * x^2+3 * x+1

· factor 函数

factor 函数的功能对于符号表达式来说是分解因式，而对一个数来说则是分解质因数，如：

》sym x; y=factor ((x^2-9))　　　返回的是 y= (x-3) * (x+3)

》factor (sym ('2730'))　　　返回的是 ans =(2)*(3)*(5)*(7)*(13)

· collect 函数

collect 函数的功能将符号表达式表示为幂的形式，如：

》syms a x; collect (x^2 * a+a * x-x^2 * a^3-2 * x)

返回的是 (a-a^3) * x^2+ (a-2) * x，表示成为 x 的幂的形式。

》syms a x; collect (x^2 * a+a * x-x^2 * a^3-2 * x, a)

返回的是 -x^2 * a^3+ (x^2+x) * a-2 * x，则表示成为 a 的幂的形式。

·horner 函数

horner 函数的功能将符号表达式表示成为嵌套的形式，如：

》sym x; horner (x^3 - 6 * x^2 + 11 * x - 6)　　返回的是 - 6 + (11 + (- 6 + x) * x) * x

·numden 函数

numden 函数的功能将符号表达式表示成为分式的形式，其用法是：

》[num, den] = numden (P); P 为一个符号表达式，输出的 num 为分子，den 为分母。

》syms x; [n, d] = numden (x/ (x - 1) - (x - 1) / (x + 1));

n = collect (n), d = collect (d)

返回的是：n = 3 * x - 1, d = x^2 - 1

· simple 函数

simple 函数的功能尽量对符号表达式进行化简，并可给出若干种形式，其用法是：

》simple (P);　　P 为一个符号表达式，输出的是各种形式的结果。

》[r, how] = simple (P);　　r 给出最简单的形式，how 给出使用的化简方式。

· subs 函数

subs 函数的功能是将已经赋值的符号变量的数值代入符号表达式，如：

》syms a x; y = (x - a) ^3; a = 2; y = subs (collect (y))

符号运算的结果是：y = x^3 - 6 * x^2 + 12 * x - 8

》syms x; y = (x - 1) * (x + 3) * (x - 4); x = 2; y = subs (y), 运算结果为 y = - 10

·solve 函数

solve 函数的功能是求解代数方程，如：

》x = solve ('a * x^2 + b * x + c');　　求解一元二次方程。

》[x, y] = solve ('a * x + b * y = e', 'c * x + d * y = f')　求解二元一次方程。

》x = solve ('cos (x)^ = x')　　求解三角方程。

·ezplot 函数

ezplot 函数是使用非常方便的函数绘图函数，其用法是：

》ezplot (f);　　绘制 f = f (x) 的图形。

》ezplot (f, xmin, xmax);　　在区间 [xmin xmax] 内绘制 f = f (x) 的图形。

4.2.2　微积分

符号运算工具箱可以进行微分、积分、级数展开、级数求和等符号运算。

· diff 函数

diff 函数的功能对符号表达式进行微分运算，其用法是：

〉diff (F);　　F 为一个符号表达式，输出的是微分的结果。
〉diff (F, n);　　F 为一个符号表达式，输出的是 n 次微分的结果。
〉diff (F, v);　　指明微分运算是针对变量 v 进行的。
〉diff (F, v, n);　　指明微分运算是针对变量 v 进行 n 次。
如:〉syms a x; diff (x^3+a*x+1);　　运算结果为 3*x^2+a
〉syms a x; diff (x^3+a*x+1, a);　　运算结果为 x
〉syms a x; diff (x^3+a*x+1, 2);　　运算结果为 6*x

· int 函数

int 函数的功能对符号表达式进行积分运算，其用法是：
〉int (F);　　F 为一个符号表达式，输出的是不定积分的结果。
〉int (F, v);　　指明不定积分运算是针对变量 v 进行的。
〉int (F, a, b);　　计算定积分，积分的下限为 a，上限为 b。
〉int (F, v, a, b);　　指明定积分运算是针对变量 v 进行的。

· limit 函数

limit 函数的功能求极限，其用法是：
〉limit (F, x, a);　　F 为一个符号表达式，输出的 x 趋于 a 时的极限，缺省值 a=0。
〉limit (F, x, a,'right');　　求右极限。
〉limit (F, x, a,'left');　　求左极限。

· taylor 函数

taylor 函数的功能是将某个函数展开为泰勒级数，其用法是：
〉taylor (F, n);　　F 为一个符号表达式，输出的是 n 项的麦克劳林级数，缺省值 n=6。
〉taylor (F, a, n);　　在 x=a 处，将 F 展开为 n 项的泰勒级数。

· symsum 函数

surf 函数的功能是进行级数求和，其用法是：
〉syms k; symsum (s, n1, n2);　　输出为级数 s_k 的和，即 $\sum_{k=n_1}^{n_2} s_k$
〉syms k n; symsum (s, n, n1, n2);　　求和针对 n 进行。
〉syms k; symsum (1/k^2, 1, inf);　　输出为 1/6*pi^2，即 $\sum_{k=1}^{\infty}\frac{1}{k^2}=\frac{\pi^2}{6}$

4.2.3 积分变换

· fourier 函数

fourier 函数的功能是进行傅立叶变换，其用法是：

》F=fourier (f);　　F=F (w) 是 f (x) 的傅立叶变换。
》F=fourier (f, u, v);　　F=F (v) 是 f (u) 的傅立叶变换。

·ifourier 函数
ifourier 函数的功能是进行傅立叶反变换，其用法是：
》f=ifourier (F);　　f=f (x) 是 F (w) 的傅立叶反变换。
》f=ifourier (F, v, u);　　f=f (u) 是 F (v) 的傅立叶反变换。

·laplace 函数
laplace 函数的功能是进行拉普拉斯变换，其用法是：
》L=laplace (f);　　L=L (s) 是 f (t) 的拉普拉斯变换。
》L=laplace (f, w, z);　　L=L (z) 是 f (w) 的拉普拉斯变换。

·ilaplace 函数
ilaplace 函数的功能是进行拉普拉斯反变换，其用法是：
》f=ilaplace (L);　　f=f (t) 是 L (s) 的拉普拉斯反变换。
》f=ilaplace (L, z, w);　　f=f (w) 是 L (z) 的拉普拉斯反变换。

· ztrans 函数
ztrans 函数的功能是进行 Z 变换，其用法是：
》F=ztrans (f);　　F=F (z) 是 f (n) 的 Z 变换。
》F=ztrans (f, k, w);　　F=F (w) 是 f (k) 的 Z 变换。

·iztrans 函数
iztrans 函数的功能是进行 Z 变换，其用法是：
》f=iztrans (F);　　f=f (n) 是 F (z) 的 Z 反变换。
》f=iztrans (F, w, k);　　f=f (k) 是 F (w) 的 Z 反变换。

4.2.4 转换

· poly2sym 函数
poly2sym 函数的功能是将系数向量转换成符号多项式，其用法是：
》y=poly2sym (v);　　将系数向量 v 转换为符号多项式 y=y (x)。
》syms t; y=poly2sym (v, t);　　将系数向量 v 转换为符号多项式 y=y (t)。

·sym2poly 函数
sym2poly 函数的功能是将符号多项式转换成系数向量，其用法是：
》v=sym2poly (f);　　将符号多项式 f (x) 转换为系数向量 v。

·char 函数

char函数的功能是将符号变量转换成字符串，其用法是：

》syms x；s=char（x）；　　　　将符号变量x转换为字符串s。

4.2.5 线性代数

符号运算工具箱可以进行线性代数的符号运算，函数的形式与数值线性代数的函数形式类似，只要变量是符号变量就进行符号运算。各种符号线性代数函数见表4.5。

表4.5 符号运算工具箱中的一些符号代数函数（矩阵为A，向量为v）

函　　数	说　　明
inv（A）	求逆矩阵
det（A）	计算行列式
rank（A）	求矩阵的秩
[B，D] =eig（A）	B矩阵的列向量为特征向量，D矩阵的对角线为特征值
diag（v），diag（A）	前者建立对角形矩阵，后者提取矩阵对角线元素
triu（A）	建立上三角矩阵
tril（A）	建立下三角矩阵
[P，J] =jordan（A）	约旦标准型 J = inv（P） * A * P
poly（A）	求特征多项式

4.2.6 演示

符号运算工具箱内设有6个专门用于演示的M函数，观看演示是掌握符号运算工具箱的一种好方法。

symintro －　符号运算工具箱的简介
symcalcdemo －　微积分符号运算的演示
symlindemo －　线性代数符号运算的演示
symvpademo －　变量精度的演示
symrotdemo －　平面旋转的演示
symeqndemo －　方程符号求解的演示

4.3 SIMULINK 工具箱

SIMULINK工具箱的可以称为系统仿真工具箱，其本意是模块分析与结构函数，实际上它是一个动态系统的仿真环境。在系统仿真工具箱内提供了许多现成的模块，使用者只需要用鼠标将相应的模块连接在一起，就可以观测到其运行结果，这样使系统的仿真过程变得相当直观，同时由于无需进行编程，因此大大地方便了使用者。由于系统仿真工具箱具备的这些优点，于是后来很多实用的工具箱都是在它的基础上编写的，具体说就是建立起特定的模块供使用者调用，这样的工具箱有：DSP（数字信号处理）工具箱、通信工具箱、电力系统工具箱、系统识别工具箱等等。

系统仿真存在着许多不同的版本，在各种版本之间存在着比较大的差异。与4.2版MATLAB配套的是1.3版，与5.2版MATLAB配套的是2.2版，与5.3版MATLAB配套的是3.0版。在本节内简要介绍2.2版的内容。

系统仿真工具箱的使用是比较简单的，同时又是比较灵活的，使用者可以根据所研究的问题，使用鼠标建立起整个的系统模型。首先进入 MATLAB 的工作环境，然后键入 simulink，于是在屏幕上就会出现 SIMULINK 的窗口，如图 4.4 所示，其中用图标的方式显示出若干现成的用于系统仿真的库（Library）。图中从左到右排列的库分别是：源模块、输出模块、离散模块、线性模块、非线性模块、接口模块，左下角是采用模块方式的其它工具箱，右下角是演示。

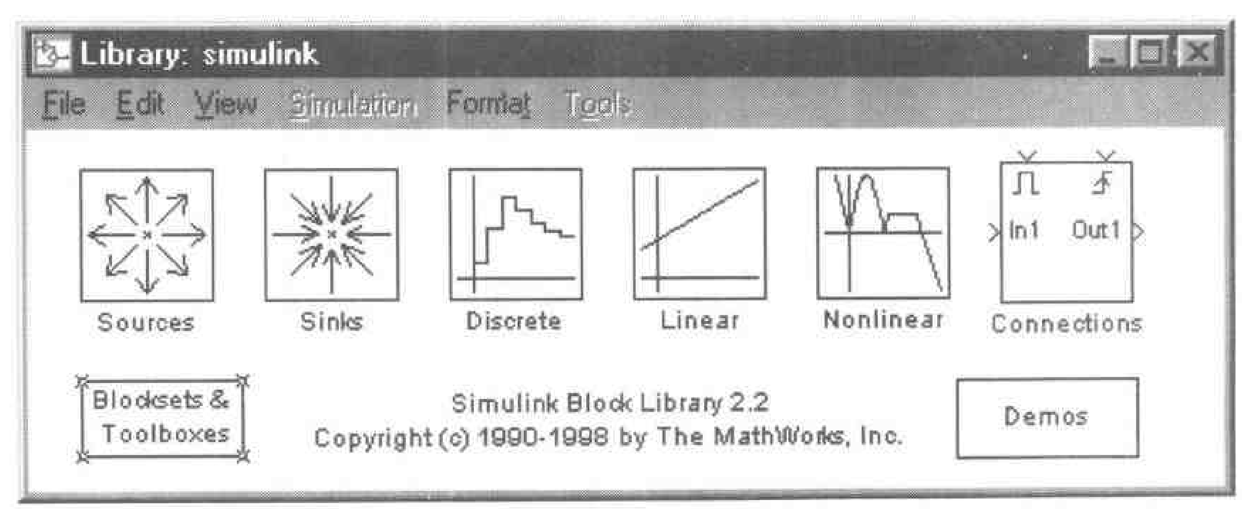


图 4.4　SIMULINK 窗口

键入 sumulink 以后，同时还打开一个名为 untitled 的窗口，使用鼠标将各模块拖进该窗口，然后将模块连接在一起就可以进行仿真了。

库内的各种模块如图 4.5 和图 4.6 所示。在源模块库内有各种信号源、噪声源和时钟；

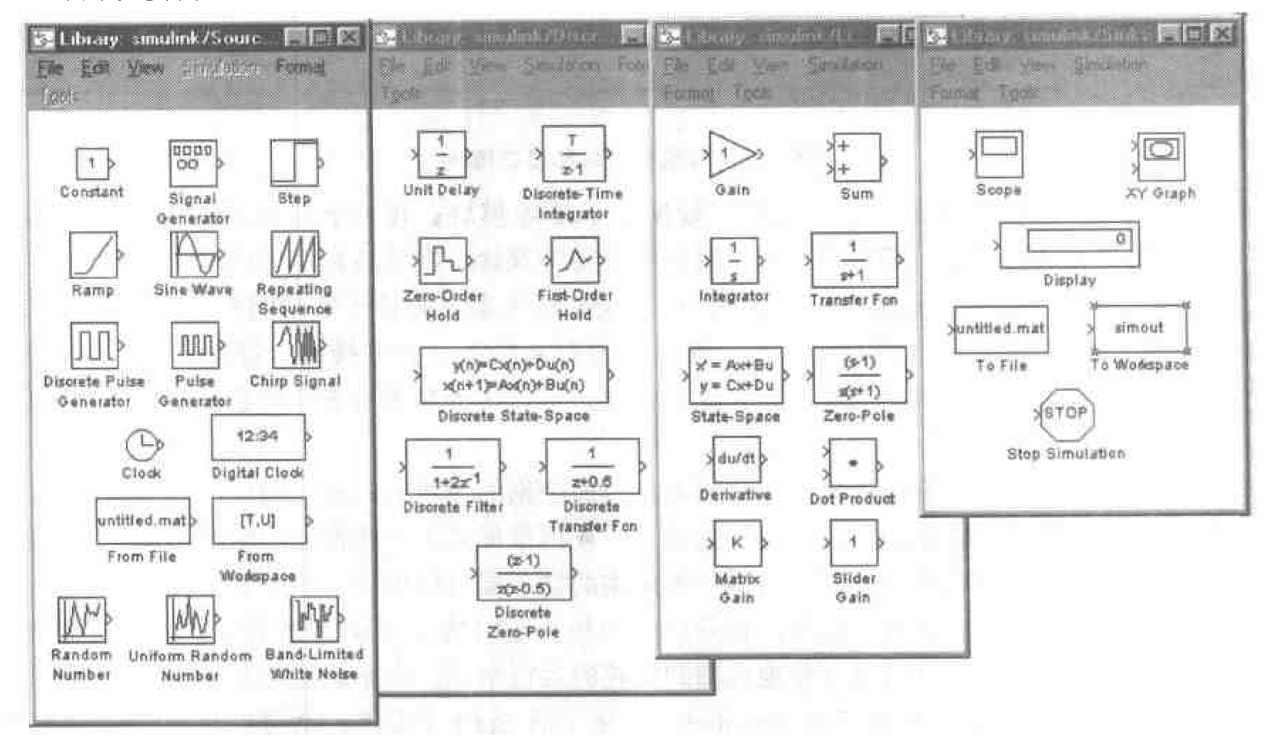


图 4.5　源模块、离散模块、线性模块和输出模块

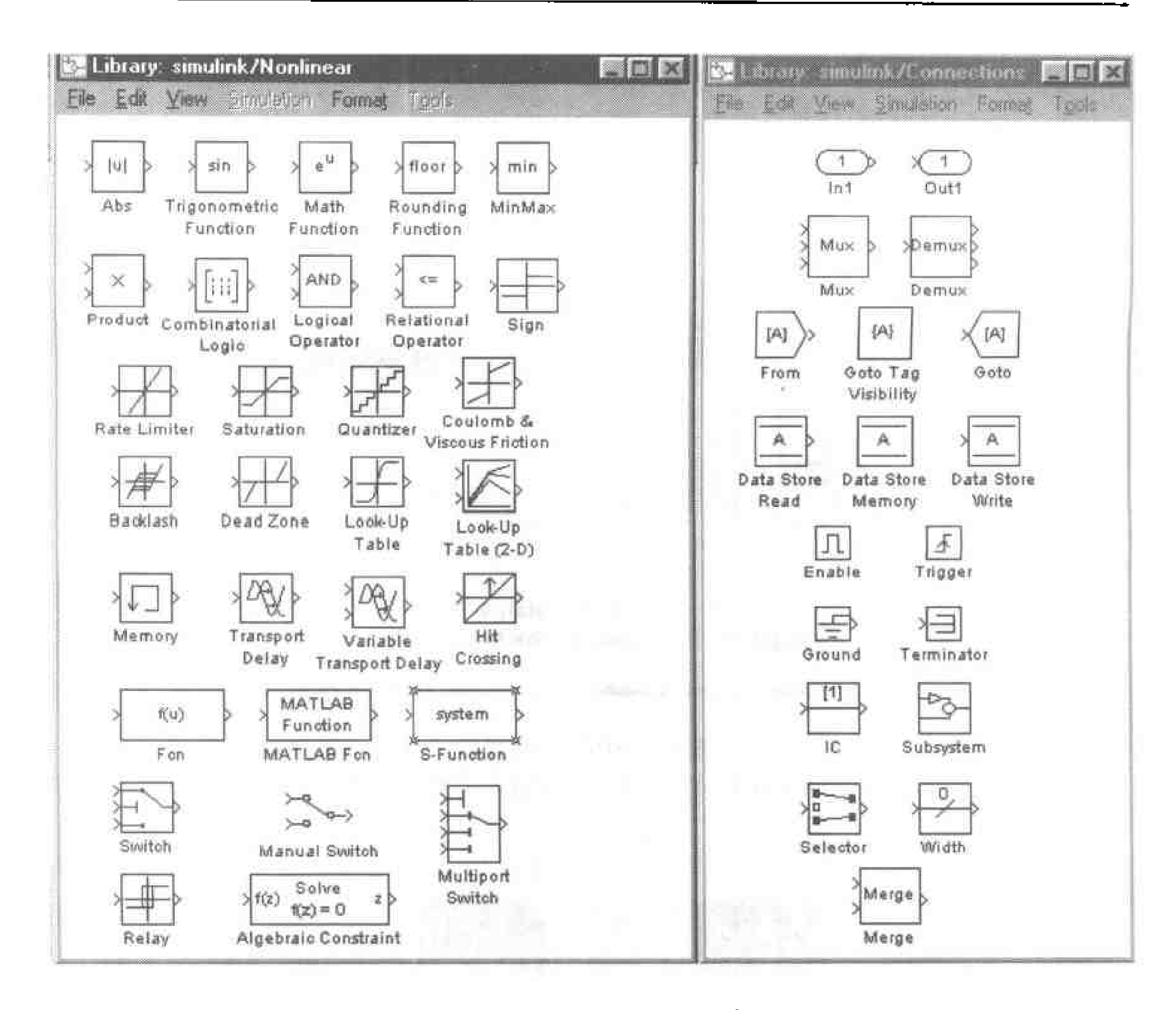


图 4.6 非线性模块和接口模块

在输出模块库内有示波器、xy 绘图仪、数据显示器等模块；在离散模块库内有单位延迟、数字滤波器、转移函数、零极点、离散时间积分器等模块；在线性模块库内有放大器、加法器、微分器、积分器、转移函数等模块；在非线性模块库内有各种数学函数、逻辑运算器、量化器、M 函数、开关等模块；在接口模块库内有复接器、去复接器、终端负载、数据存储器、子系统等模块。灵活地使用上述这些模块，可以实现对相当复杂的系统进行计算机仿真。

多数的模块同时具有输入端口和输出端口，而个别的模块只有输入端口，如示波器，而另外一些模块则仅有输出端口，如信号发生器。使用鼠标点击一些模块的输出端口，然后按着鼠标进行移动，就可将一个模块与另一个模块的输入端口连接在一起。数个模块合理地连接在一起就构成了一个完整的系统，在系统中一些关键的地点安装上显示设备，如示波器、绘图仪、数据显示器，就可以方便地观测到系统的运行结果。当系统建立起来以后，用鼠标点击菜单栏上的 Start 图标或点击 Simulation 菜单下的 Start 子菜单，就可以开始进行系统仿真。

很多模块的参数是可以改动的。改动的方法是使用鼠标点击某个模块两次，这样就打开了一个对话框，在对话框内便可以对该模块的参数进行修改。

例 1　信号发生器和示波器

通过这个例子，读者对于系统仿真工具箱的使用会有一个比较具体的认识。首先从源模块库内拖出 3 个信号发生器模块，然后分别对它们进行设置，一个为噪声源，另外两个为正弦波，频率分别为 0.2Hz 和 0.6Hz；接着从线性模块库内拖出一个加法器，并将它的输入端设置为 3 个；最后再从输出模块库内拖出一个示波器。将上述的 5 个模块连接在一起，用鼠标点击菜单栏上的 Start 图标就可以进行系统仿真了，在示波器的图标上点击两下，就可以看到仿真的结果了，见图 4.7。

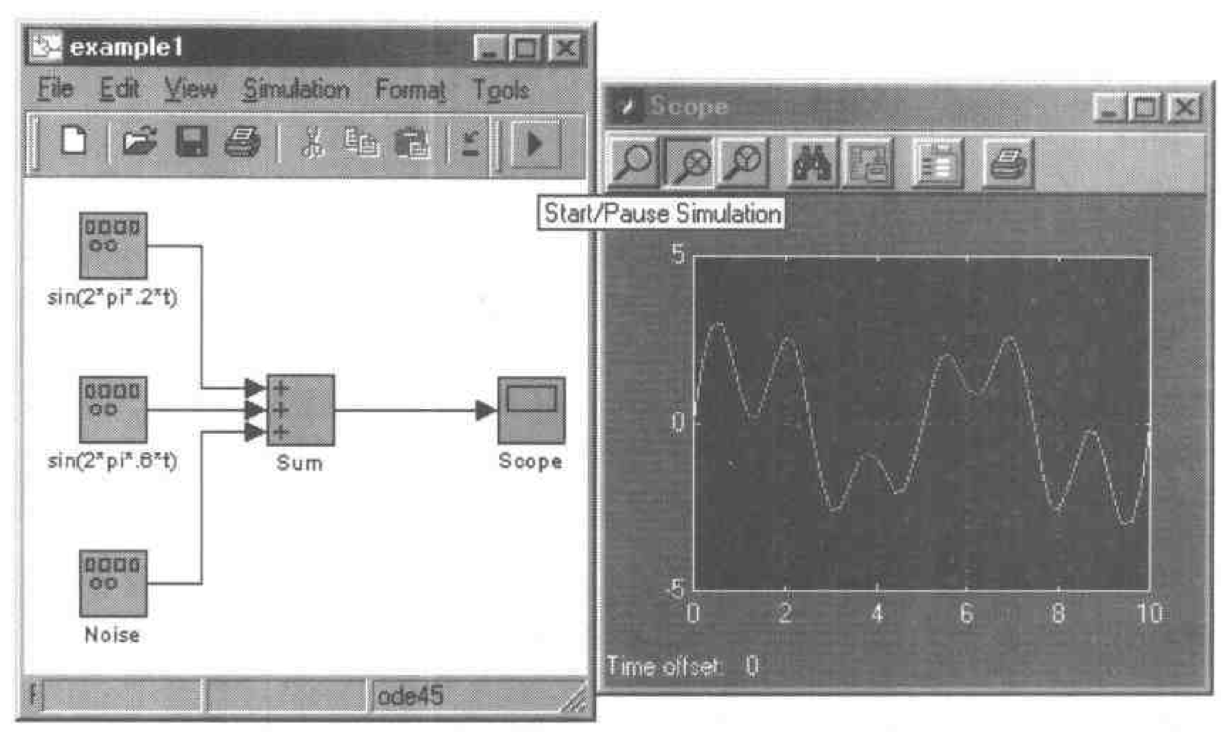


图 4.7　信号发生器和示波器

用鼠标点击 File 菜单内的 Save 子菜单，将整个系统储存在一个名为 example1.mdl 的文件内。在 MATLAB 工作环境下，键入 example1 就可以重现上述的系统仿真过程。

使用者可以构造自己的模块，其方法是：使用鼠标将所要封装的子系统选中，然后点击 Edit 菜单内的 Create Subsystem 菜单便可构造一个自定义的模块。

在 SIMULINK 环境中，还可以使用函数的方式来改变各模块的参数，在表 4.6 中列出了一些模块构造函数。

表 4.6　系统仿真工具箱的模块构造函数

函　数	说　明
find-system	寻找模块系统
set-param	设置模块参数（与 set 函数相近）
get-param	获取模块参数（与 get 函数相近）
gcb	获取当前模块的名称
gcs	获取当前系统的名称

在SIMULINK环境中，还可以使用S函数的方式来建立起自定义的微分方程、离散系统方程以及相关的运算方式，S函数对应着一个自定义的系统模块。关于S函数的句法，键入help sfuntmpl便可得到详细的联机帮助文件。

在SIMULINK中设置了若干很精巧的显示程序，比较全面的展示出了SIMULINK的各种基本功能，同时深入地了解演示的具体环节也是掌握SIMULINK的好方法。

4.4 DSP工具箱

DSP工具箱的全称为数字信号处理工具箱，它实际上是由一些扩充的SIMULINK库组成的，库中的模块是专门为了进行数字信号处理而设计的，也就是说DSP工具箱实际上是SIMULINK的扩展或子集。DSP工具箱也存在着不同的版本，与4.2版MATLAB配套的是1.0a版，与5.2版MATLAB配套的是2.2版，与5.3版MATLAB配套的是3.0版。在本节内，简明扼要地介绍2.2版的主要内容。

为了适应数字信号处理的需要，在DSP工具箱中主要增加了三个功能：1）缓存技术，2）支持复数，3）统计函数。

在DSP工具箱内使用缓存器将连续的时间序列变成了若干并行的列向量，缓存技术的使用是DSP工具箱的核心内容之一，这样便于对信号进行各种处理，如使用FFT来进行谱分析。由于信号通常是用复数来表示的，因此在DSP工具箱内增添了许多进行复数运算的模块。同时为了分析随机信号，在DSP工具箱内还增添了若干统计模块。DSP工具箱与信号处理工具箱的联系也是是很紧密的，其内部要调用许多信号处理工具箱内的M函数，两者的区别是，信号处理工具箱采用编写M函数的方式来进行信号处理，而DSP工具箱是使用模块的方式来完成相应的工作。

键入dsplib，屏幕上就会出现DSP的模块库，如图4.8所示。从左向右排列的依次是：DSP源模块库、DSP输出模块库、DSP通用模块库、数学函数模块库、滤波器模块库、谱分析模块库和显示。使用鼠标点击模块库便可进一步显示出各种DSP模块。DSP源模块包括赋值（常数、向量、矩阵）、窗函数等模块；DSP输出模块包括时间示波器、频谱仪、数据输出（常数、向量、矩阵）等模块；DSP通用模块包括延迟、变换、缓存器、开关等模块；滤波器模块包括滤波器、滤波器实现、有源滤波器等模块；谱分析模块中有三种谱分析方法。使用这些模块可以建立起相当真实的数字信号系统，进一步可以对该系统进行仿真，以得到各种参数。

例2 信号发生器和示波器

DSP工具箱的使用方法与SIMULINK是一样的。首先，从SIMULINK的源模块库内找到信号发生器模块，从SIMULINK的离散模块库内找到采样模块，从SIMULINK的接口模块库内找到复接器模块，从SIMULINK的输出模块库内找到示波器模块；从DSP输出模块库内找到带缓存的FFT示波器（频谱仪），从DSP滤波器模块库内找到IIR滤波器设计模块，然后按照图4.9中所示的方式连接在一起。使用鼠标的右键点击模块，可将各个模块设置成不同的颜色，以示区别。

设定信号发生器的参数为：信号形式为方波信号、频率为100Hz、幅度为2。在采样模块内设定采样时间间隔为0.00005s，也就是采样频率为20kHz。设定IIR滤波器的参数为：4阶的切比雪夫II型低通滤波器、通带的上限频率的归一值为0.02（即200Hz），阻带内的

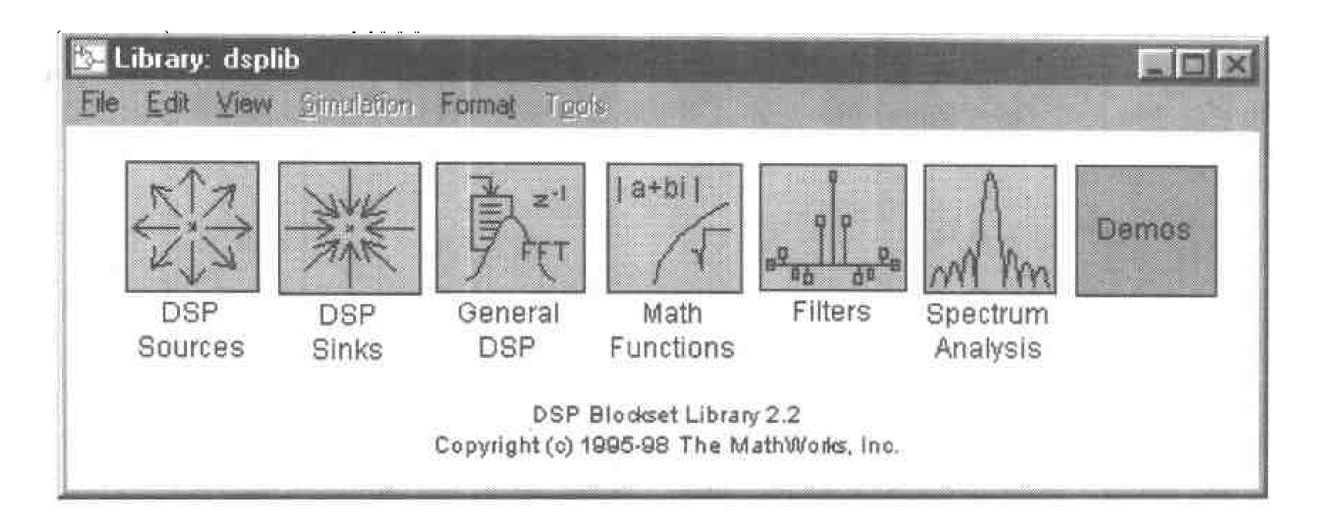


图 4.8 DSP 模块库

图 4.9 方波信号与数字滤波器

最小衰减为 40dB。设定 FFT 示波器的采样时间间隔为 0.0005s，即采样频率为 2kHz，而其它的参数均使用缺省参数。最后将当前的系统存在 exampl2.mdl 文件中。

点击菜单栏中的 Start 菜单就可以进行仿真了，仿真的结果见图 4.9，由于在系统中使用了复接器，因此示波器变成为双踪示波器。从仿真结果中我们可以清楚地看出，方波的频

率成分有 100、300、500、700、900Hz 等等，而在滤波器之后，信号中就只剩下了 100Hz 的正弦波（品红色曲线），同时还可以看出滤波器的相移对信号产生的时间滞后现象。这个例子清楚地表明了滤波器的作用。另外，还可以将信号方式设置为锯齿波或正弦波，然后再观察滤波的效果。

4.5 COMM 工具箱

COMM 工具箱（即通信工具箱），它专门用来解决通信领域内的各种问题。通信工具箱内设置了许多 SIMULINK 环境下的专用模块，同时还设置了若干与上述模块对应的 M 函数，这样分析通信领域内的问题即可以使用系统仿真的方式来进行，也可以使用编程的方式来进行，给使用者提供了很大的方便。通信工具箱也有若干版本，与 4.2 版 MATLAB 配套的是 1.0a 版，与 5.2 版 MATLAB 配套的是 1.3 版，与 5.3 版 MATLAB 配套的是 1.4 版，各版本之间差别不大。在本节内，简明扼要地介绍 1.3 版的主要内容。

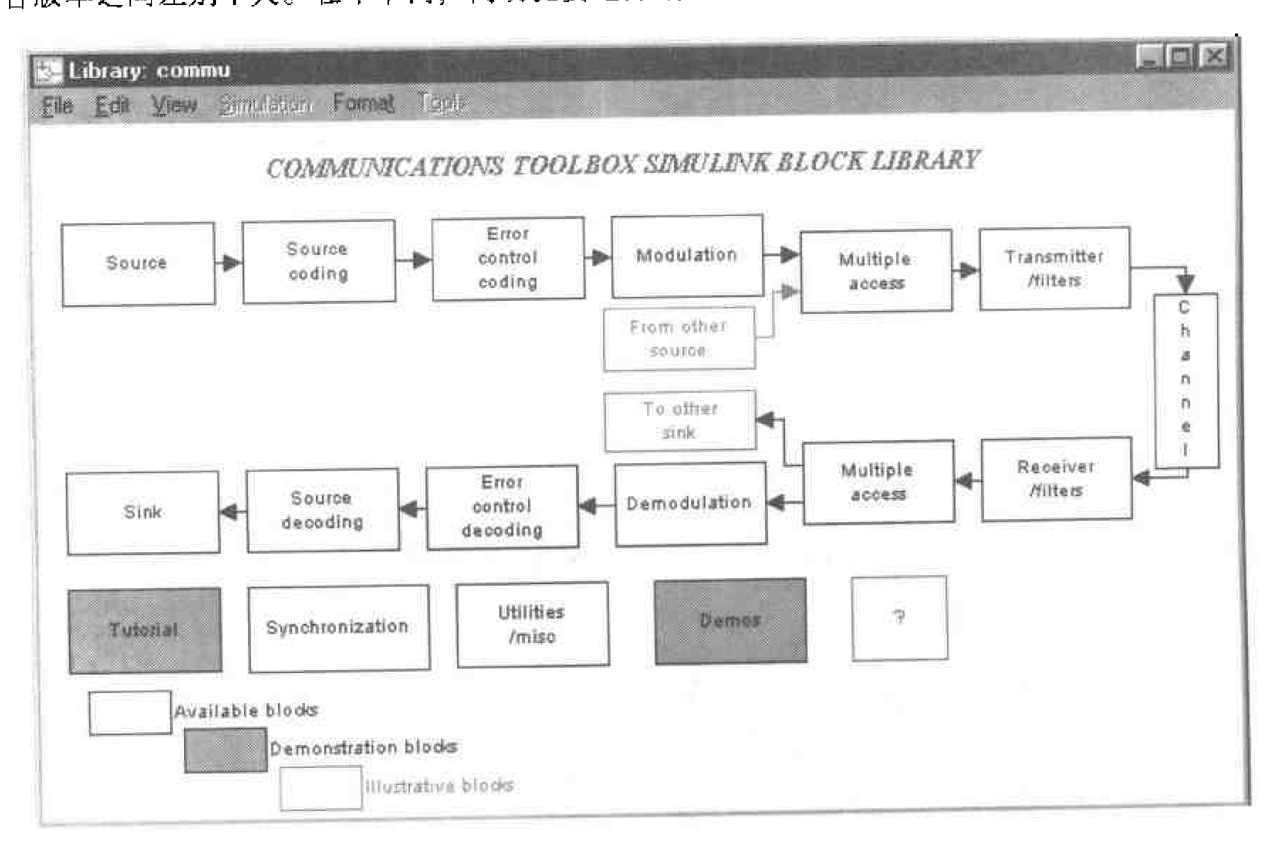


图 4.10 通信工具箱的模块库

键入 commlib，屏幕上就会出现通信工具箱的模块库，它们是按照一个完整的通信系统框图来排列的，如源编码模块库、差错控制编码模块库、调制解调模块库、信道模块库等等，另外附加上同步模块库、辅助功能模块库、演示部分和示教部分，如图 4.10 所示。对于大部分的模块，在通信工具箱内还设置了相应的 M 函数。

显然使用通信工具箱对电子系统进行仿真是十分便利的，使用该工具箱可以加强对通信系统和通信原理的理解，同时还可以分析一些具体的工程问题。

4.5.1 信号源与显示

信号源与显示部分提供的 M 函数见表 4.8；同时还提供了许多模块和相应的演示模型，其中包括各种信号发生器、各种噪声发生器、误码仪、眼图绘图仪等等，模块的数量多于 M 函数的数量。

表 4.8 信号源与显示部分的 M 函数

函　　数	说　　明
randint(m, n)	随机二进制信号发生器(m×n 矩阵)
randint(m, n, [i1 i2])	随机整数信号发生器(m×n 矩阵)
randbit(m, n)	随机二进制噪声发生器(m×n 矩阵)
[num, rat] = biterr(x, y)	计算比特误差率(误信率), num 为比特误差的数量, rat 为比特误差率
[num, rat] = symerr(x, y)	计算误码率, num 为码元误差的数量, rat 为误码率
eyescat(x, fd, fs)	绘制眼图, fs 为采样频率, fs/(2 * fd)为正整数

4.5.2 源编码

源编码部分提供的 M 函数见表 4.9；同时还提供了若干编解码模块和相应的演示模型，其中包括量化器和逆量化器、μ 律压缩器和扩张器、A 律压缩器和扩张器、DCPM 编码器和解码器、触发量化器等。

表 4.9 源编码部分的 M 函数

函　　数	说　　明
y = compand (x, mu, v)	μ 律压缩器，mu 为 μ 参数，v 为信号的最大值
y = compand (x, par, v, method)	μ 律或 A 律压缩器和扩张器，par 为 μ 或 A 参数，v 为信号的最大值，字符串 method 用来说明具体的工作方式，如 'A/expander'代表 A 律扩张器
dpcmdeco	DCPM（增量脉冲编码调制）解码器
dpcmenco	DCPM 编码器
dpcmopt	DCPM 参数优化设计
lloyds	使用训练数据对标量量化进行优化设计
quantiz	量化器和量化指数计算

4.5.3 差错控制编码

差错控制部分提供的 M 函数见表 4.10；同时还提供了若干编解码模块和相应的演示模型，其中包括汉明码、线性码、RS 码、循环码、BCH 码等。

表 4.10　差错控制编码部分的 M 函数

函　　数	说　　明
code = encode（msg，N，K，method，opt）	差错控制码编码器，code 为编码器输出，msg 为信息，N 为码元的长度，K 为信息的长度，method 用来说明编码的方式，opt 为具体编码方式的参数，
code = decode（msg，N，K，method）	差错控制码解码器

M 函数支持的编码方式有：

'hamming' – 汉明码　　'linear' – 线性码　　'cyclic' – 循环码

'bch' – BCH 码　　'rs' – RS 码　　'convol' – 卷积码

4.5.4　调制与解调

调制与解调部分提供的 M 函数见表 4.11；同时还提供了若干调制/解调模块和相应的演示模型，其中包括通带和基带的调幅、调相、调频、QASK、QPSK、MASK、MFSK、MPSK 等方式。

表 4.11　调制与解调部分的 M 函数

函　　数	说　　明
y = amod（x，fc，fs，method）	模拟调制，y 为模拟调制器输出，x 为调制信号，fc 为载波频率，fs 为采样频率，字符串 method 用来说明模拟调制的方式。
x = ademod（y，fc，fs，method）	模拟解调器
y = amodce（x，fs，method）	模拟调制的复包络
x = ademodce（y，fs，method）	从复包络中解调出调制信号
y = dmodce（x，fd，fs，method）	数字调制的复包络
x = ddemodce（x，fd，fs，method）	从复包络中解调出数字调制信号
y = modmap（x，fd，fs，method）	数字调制信号与模拟信号的映射
x = demodmap（y，fd，fs，method）	数字调制信号解调的逆映射

M 函数支持的模拟调制方式有：

'am' – 调幅　　'amdsb – sc' – 双边带调幅　　'amssb' – 单边带调幅

'qam' – 正交调幅　'fm' – 调频　　'pm' – 调相

M 函数支持的数字调制方式有：

'ask' – ASK（幅度键控）　　'psk' – PSK（相移键控）

'qask' – QASK（正交幅度键控）　'fsk' – FSK（频移键控）

4.5.5　多址连接

多址连接部分支持 CDMA（码分多址）、FDMA（频分多址）和 TDMA（时分多 址）三种方式，但是没有提供相应的 M 函数。

4.5.6 发送与接收滤波器

发送与接收滤波器部分提供的 M 函数见表 4.12；同时还提供了若干滤波器模块和相应的演示模型，其中包括升余弦滚降滤波器、希尔伯特滤波器、SINC 滤波器等。

表 4.12 发送与接收滤波器部分的 M 函数

函数	说明
[num, den] = hank2sys (H)	将汉克尔矩阵 H 变换成线性系统的转移函数，num 为该函数的分子系数向量，den 为该函数的分母系数向量
[b, a] = hilbiir (st)	设计希尔伯特 IIR 滤波器
[num, den] = imp2sys (imp)	根据冲击响应来确定线性系统的转移函数
y = rcosflt (x, fd, fs)	使用升余弦滚降滤波器进行滤波
b = rcosine (fd, fs)	升余弦滚降滤波器的设计

4.5.7 信道

信道部分提供了若干基带信道和通带信道的模块，如瑞利信道、加性白高斯噪声信道等，但未提供相应的 M 函数。

4.5.8 同步

同步部分提供了四种锁相环模块和一个演示，但未提供相应的 M 函数。

4.5.9 辅助功能

辅助功能部分提供了若干辅助模块，同时还提供了 M 函数，见表 4.13。

表 4.13 辅助功部分的 M 函数

函数	说明
d = bi2de (b)	将二进制向量转换为十进制向量
b = de2bi (d)	将十进制向量转换为二进制向量
checkinp	查询 M 函数的缺省值
A = vec2mat (v, n)	将向量 v 按行转换成为 n 列的矩阵，其余的元素为 0
blkdiag (A, B, C)	生成块对角矩阵

4.6 COMPILER 工具箱

COMPILER 工具箱就是编译工具箱，其功能是将 MATLAB 的 M 函数转换成为 C 语言或 C++ 语言的源程序，同时还可以对源程序进行编译生成可执行文件，这样完成 M 函数的功能就完全可以脱离 MATLAB 环境而在 WINDOWS 环境下执行了，于是仿真程序就成为了一种真正意义上的实用程序。

编译工具箱有若干版本，与 4.2 版 MATLAB 配套的是 1.0 版，与 5.2 版 MATLAB 配

套的是1.2版，与5.3版MATLAB配套的是2.0版，各版本之间差别较大。在本节介绍1.2版编译工具箱的使用。

使用mcc命令就可将MATLAB的M函数转换成为C语言或C++语言的源程序，另外还可以将C语言编写的程序转换成为在MATLAB中可以直接调用的MEX文件，其基本用法是：

》mcc [-options] fun [fun2 ……]

mcc命令各参数的主要功能为：

将fun.m转换成为fun.c或fun.cpp，若后面增加另外一个或数个M函数的名称，则将这些M函数同时进行转换；将fun.c转换成为MEX文件。将转换后的文件储存在当前的目录下。

OPTIONS：用来指定mcc命令的具体功能

-c　将M函数仅仅转换为C语言源程序，不生成MEX文件或独立应用程序。

-e　将M函数转换为C文件，并与MATLAB C数学库连接在一起，这样就可以脱离MATLAB的环境而运行。当M函数的名称为mai.m时，生成独立应用程序。

-g　调试，包括了许多调试符号的说明。

-h　同时对帮助函数进行编译。

-l　运行错误时，显示M函数的行数。

-m　将main.m函数转换为C语言的main函数，既没有输入参数，也没有输出参数，生成可独立使用的应用程序。

-p　将M函数转换为C++文件，并与MATLAB C++数学库连接在一起，这样就可以脱离MATLAB的环境面运行。当M函数的名称为main.m时，生成独立应用程序。只可与m、g、v参数同时使用。

-q　快速模式。

-r　实数方式，输入参数和输出参数均为实数。

-s　给全程变量指定静态存储区域。

-S　SIMULINK的C-MEX S函数模式。

-t　在C原程序内生成轨迹打印语句。

-u<number>　生成SIMULINK兼容的C-MEX S函数，并且指定输入端口的数量为<number>.。

-v　显示编译的过程。

-wc　打印警告信息，其中包括哪个程序段的运行速度较慢、M函数与转换后的C语言源程序之间的区别。

-y<number>　生成SIMULINK兼容的C-MEX S函数，并且指定输出端口的数量为<number>.。

-z<path>　指定库的使用路径。

实例：

mcc -c specanal.m　生成specanal.c程序

参考文献

[1] (美) Duane Hanselman, Bruce Littlefield. 精通 MATLAB. 西安：西安交通大学出版社，1988
[2] 薛定宇．控制系统计算机辅助设计．北京：清华大学出版社，1996
[3] (美) 帕普里斯 A．信号分析．北京：科学出版社，1981
[4] 胡广书．数字信号处理．北京：清华大学出版社，1997
[5] (美) 英格尔 V, 普罗克斯 J. 数字信号处理及其 MATLAB 实现．北京：电子工业出版社，1998
[6] 樊昌信，詹道庸，徐柄祥等．通信原理．北京：国防工业出版社，1995

第五章　电路、网络与系统

在本章内通过列举实例的方式来介绍如何使用 MATLAB 对电路、网络和系统进行仿真的问题，这些实例均是由本书作者自己编写的。通过前面四章的介绍，读者目前手头已经拥有了充分的工具和手段来对各种无源网络和有源网络的特性参数进行细致的分析，对网络时域内特性参数的分析是依靠求解微分方程来解决的，而对网络频域内特性参数的分析则是采用转移函数的方式来完成的。

本章列举的各种实例基本上是使用 5.2 版 MATLAB 来进行的，因为涉及符号运算问题时采用 5.2 版的 MATLAB 比较方便。

5.1　两端口网络

两端口网络通常又称为双口网络，它有一个输入端口和一个输出端口，是一种最基本的网络形式，例如滤波器、衰减器、均衡器、音量控制器、音调控制器都是常见的两端口网络。分析两端口网络通常是在频域内进行的，因为人们关心的主要是该网络的转移函数（传递函数）、输入阻抗、输出阻抗等参数。

5.1.1　网络参数

两端口网络的框图见图 5.1，分析两端口网络通常可以使用 Z 参数、Y 参数、A 参数、H 参数和 G 参数，各种参数用矩阵方式可表示为：

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix},$$

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix},$$

$$\begin{bmatrix} U_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} U_2 \\ -I_2 \end{bmatrix}$$

$$\begin{bmatrix} U_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ U_2 \end{bmatrix}, \quad \begin{bmatrix} I_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ I_2 \end{bmatrix},$$

图 5.1　两端口网络

其中，Z 矩阵与 Y 矩阵互为逆矩阵，H 矩阵和 G 矩阵互为逆矩阵，而 A 矩阵与 Z 矩阵、A 矩阵与 H 矩阵的关系分别为：

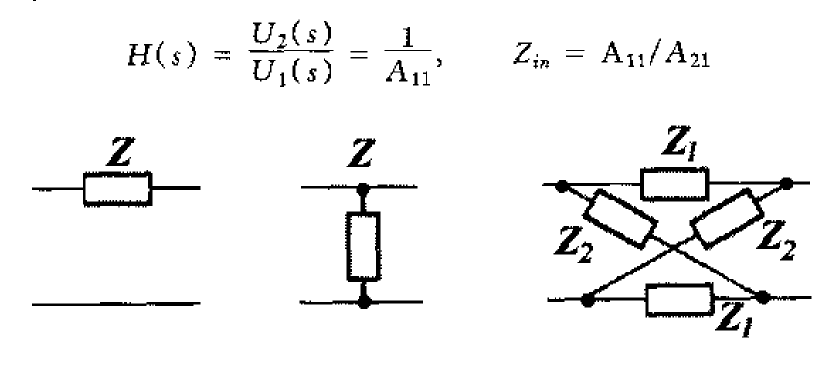
$$[A] = \frac{1}{Z_{21}} \begin{bmatrix} Z_{11} & |Z| \\ 1 & Z_{22} \end{bmatrix}, \qquad [Z] = \frac{1}{A_{21}} \begin{bmatrix} A_{11} & |A| \\ 1 & A_{22} \end{bmatrix},$$

$$[A] = \frac{-1}{H_{21}} \begin{bmatrix} |H| & H_{11} \\ H_{22} & 1 \end{bmatrix}, \qquad [H] = \frac{1}{A_{22}} \begin{bmatrix} A_{12} & |A| \\ -1 & A_{21} \end{bmatrix}$$

当两个两端口网络进行级联时（即第一个网络的输出端与第二个网络的输入端直接连接在一起），复合网络的A矩阵可由两个网络A矩阵相乘而得到，因此对两端口网络进行分析时使用A矩阵较为方便，而Z矩阵和H矩阵则用于分析两个两端口网络串联和串并联的情况。

当2端口开路（未接负载）时，$I_2=0$，于是两端口网络的转移函数（传递函数）、输入阻抗分别可表示为：

$$H(s)=\frac{U_2(s)}{U_1(s)}=\frac{1}{A_{11}},\qquad Z_{in}=A_{11}/A_{21}$$



（a）串联阻抗　　（b）并联阻抗　　（c）X型网络

图5.2　基本的两端口网络

基本的两端口网络如图5.2所示，它们的A矩阵分别为：

$$\text{(a)}\ [A]=\begin{bmatrix}1 & Z\\0 & 1\end{bmatrix},\qquad \text{(b)}\ [A]=\begin{bmatrix}1 & 0\\1/Z & 1\end{bmatrix},\qquad \text{(c)}\ [A]=\frac{1}{Z_2-Z_1}\begin{bmatrix}Z_1+Z_2 & 2Z_1Z_2\\2 & Z_1+Z_2\end{bmatrix}$$

由于通常的两端口网络大多可以看作是串联阻抗和并联阻抗级联而成的（显然X型网络是个例外），因此网络仿真的问题就变成了基本A矩阵连乘的问题，在MATLAB中进行这样的计算是非常容易的，A矩阵计算出来之后，转移函数和输入阻抗也就得到了。

5.1.2　T－Π变换

T形网络和Π型网络也是两种基本的网络形式，见图5.3。在实际工作中经常需要将T形网络变换为Π型网络，或将Π型网络变换成T形网络，这种变换又称为星型－三角形变换。变换的公式为：

T→Π：　$Z_a=B/Z_3$，$Z_b=B/Z_2$，$Z_c=B/Z_1$，　其中 $B=Z_1Z_2+Z_2Z_3+Z_3Z_1$

Π→T：　$Z_1=Z_aZ_b/Z_T$，　$Z_2=Z_aZ_c/Z_T$，　$Z_3=Z_bZ_c/Z_T$

其中 $Z_T=Z_a+Z_b+Z_c$

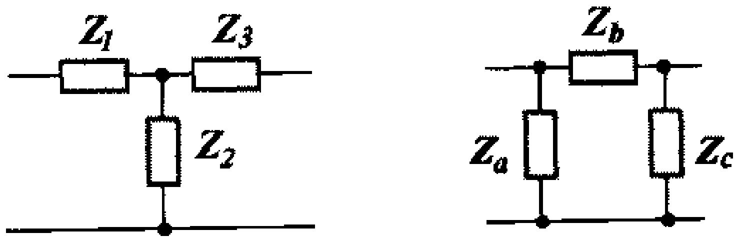


图5.3　T形网络和Π型网络

此处给出两个自定义函数 pi2tee.m 和 tee2pi.m，它们用来解决 T－Π 变换的问题。若在 5.2 版内使用这两个函数，在使用缺省输入参数的情况下还可以给出上述变换的符号表达式。

```
function [z1, z2, z3] =pi2tee (za, zb, zc);
%
%    Usage: [z1, z2, z3] =pi2tee (za, zb, zc);    BBI   2000
v=version;
if strcmp (v (1),'5');
  cmd='syms za zb zc;';                    % MATLAB 5.x
else;
  cmd='za=75; zb=za; zc=za;';   % MATLAB 4.2
end;
if nargin<1;
  eval (cmd);
else;
  if nargin<2; zb=za;        end;
  if nargin<3; zc=za;        end;
end;
z=za+zb+zc;
z1=za*zb/z;        z2=za*zc/z;        z3=zb*zc/z;
if nargout<1; [z1; z2; z3], end;
```

```
function [za, zb, zc] =tee2pi (z1, z2, z3);
%
%   Usage: [za, zb, zc] =tee2pi (z1, z2, z3);    BBI   2000
v=version;
if strcmp (v (1),'5');
  cmd='syms z1 z2 z3;';                    % MATLAB 5.x
else;
  cmd='z1=75; z2=z1; z3=z1;';   % MATLAB 4.2
end;
if nargin<1;
  eval (cmd);
else;
  if nargin<2; z2=z1;        end;
  if nargin<3; z3=z1;        end;
```

```
end;
z=z1*z2+z2*z3+z3*z1;
za=z/z3;      zb=z/z2;      zc=z/z1;
if nargout<1; [za; zb; zc], end;
```

5.1.3 两端口网络的连接

两个两端口网络的连接方式有：串联、并联、串并联、并串联和级联，连接之后形成一个复合网络。设一个网络为 a，另一个网络为 b。

· 串联

两个两端口网络串联是将两个输入端口和两个输出端口分别串联起来，串联之后相应端口的电压相加，而电流相等。对应于串联方式，复合网络的 Z 矩阵为两个 Z 矩阵之和，即 $Z= Z_a+Z_b$。

·并联

两个两端口网络并联是将两个输入端口和两个输出端口分别并联起来，并联之后相应端口的电流相加，而电压相等。对应于并联方式，复合网络的 Y 矩阵为两个 Y 矩阵之和，即 $Y= Y_a+Y_b$。

·串并联

两个两端口网络串并联是将两个输入端口串联起来，将而两个输出端口并联起来，这样输入端口是电压相加，而输出端口是电流相加。对应于串并联方式，复合网络的 H 矩阵为两个 H 矩阵之和，即 $H=H_a+H_b$。

·并串联

两个两端口网络并串联是将两个输入端口并联起来，将而两个输出端口串联起来，这样输入端口是电流相加，而输出端口是电压相加。对应于并串联方式，复合网络的 G 矩阵为两个 G 矩阵之和，即 $G=G_a+G_b$。

·级联（链接）

两个两端口网络级联是将前面网络的输出端口直接与后面网络的输入端口连接在一起，这是一种最为普遍的连接形式。对应于级联方式，复合网络的 A 矩阵为两个 A 矩阵之乘积，即 $A=A_a A_b$。

例 1　双 T 选频网络

双 T 选频网络结构见图 5.4，它由两个 T 形网络并联而成，其特点是在某个频率点上 f_0，输出为零。参数 α 和 β 关系为：$\beta=\frac{\alpha}{1+\alpha}$;。转移函数和输入阻抗表达式为：

$$H=\frac{\text{alfa}\ (1+S^2C^2R^2)}{S^2C^2R^2\ \text{alfa}+2CR\ (1+\text{alfa})\ s+\text{alfa}}$$

$$Z_{in}=\frac{S^2C^2R^2+2CR\ (1+alfa)\ s+alfa}{2sC\ (1+alfa)\ (1+sCR)}$$

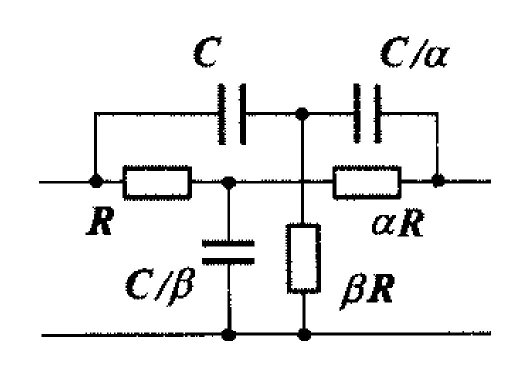


图 5.4 双 T 选频网络

分析网络问题最好使用符号运算的方式，因此应该使用 5.2 版的 MATLAB 来进行网络仿真。根据双 T 选频网络的结构，分析它的特性可以采用两个导纳矩阵相加的方式，由于选频网络的负载阻抗通常都比较大，因此可以假设网络的输出端是开路的，这样突出了问题的主要部分，同时与实际情况也是吻合的。具体的步骤是：1）分别计算两个 T 形网络的 A 参数，2）将 A 参数转换成为 Y 参数，3）将两个 Y 矩阵相加，得到双 T 选频网络的 Y 参数，4）将 Y 参数转换成为 A 参数；5）由 A 参数求出网络的转移函数，进而得到其幅频特性和相频特性，6）由 A 参数求出网络的输入阻抗。

自定义 M 函数 doubleT.m 就是专门用来分析双 T 选频网络特性的，它只能在 5.x 版下运行，其用法是：

》doubleT; 采用符号运算方式分析双 T 选频网络特性，给出该网络的幅频特性曲线，同时给出转移函数和输入阻抗的表达式。

》[B, A, Zin] = doubleT; B、A 和 Zin 均为符号表达式，B 是转移函数的分子，而 A 是转移函数的分母，Zin 为输入阻抗。

》doubleT (alfa, R, C); 根据输入参数，给出该网络的幅频特性曲线和相频特性曲线。

程序清单如下：

```
function [B, A, Zin] =doublet (alfa, R, C);
%
%       BBI 2000
if nargin<1;    syms R C alfa;
else;
  if nargin<2;    R=1e+04;                end;
  if nargin<3;    C=.015915*1e-6;         end;
end;
syms s;    beta=alfa/ (1+alfa);
A1= [1 R; 0 1] * [1 0; s*C/beta 1] * [1 alfa*R; 0 1];          % 计算A矩阵
A2= [1 1/ (s*C); 0 1] * [1 0; 1/ (beta*R) 1] * [1 alfa/ (s*C); 0 1];
Z1= [A1 (1, 1) det (A1); 1 A1 (2, 2)] /A1 (2, 1);        % A矩阵变换为Z矩阵
Z2= [A2 (1, 1) det (A2); 1 A2 (2, 2)] /A2 (2, 1);
Y=inv (Z1) +inv (Z2);        Z=inv (Y);        %网络并联，Y矩阵相加
A= [Z (1, 1) det (Z); 1 Z (2, 2)] /Z (2, 1);        % Z矩阵变换为A矩阵
H=1/A (1, 1);    Zin=A (1, 1) /A (2, 1);          % 转移函数和输入阻抗
```

```
H=simplify (H); [B, A] =numden (H); B=collect (B); A=collect (A);
[B, b0] =s_poly (B);    A=s_poly (A);
Zin=simplify (Zin); [b, a] =numden (Zin);      b=collect (b);
a=collect (a); a=factor (a); b=s_poly (b);    Zin=b/a;
if nargout<1;
  if nargin<1;             % 给出符号运算的结果
  format compact; H=' ', pretty (factor (B) /A),
  Zin=' ', pretty (b/a),
  alfa=1; R=1; C=1; B1=subs (B); B1=sym2poly (B1);
  A1=subs (A); A1=sym2poly (A1);
  w=logspace (-2, 2, 101); H=freqs (B1, A1, w);
  semilogx (w, abs (H),'b'); hold on;
  text (12, .85, ['a = ' num2str (alfa)],'horiz','left', …
    'FontName','symbol');
  alfa=100; B1=subs (B); B1=sym2poly (B1);
  A1=subs (A); A1=sym2poly (A1);
  H=freqs (B1, A1, w); semilogx (w, abs (H),'b:');
  grid; hold off;
  text (4, .95, ['a = ' num2str (alfa)],'horiz','right', …
    'FontName','symbol');
  set (gca,'FontAngle','italic','FontName','Times');
  xlabel ('f / fo'); ylabel ('abs (H)'); format loose;
  else;                    %根据输入参数，给出运算的结果
  B=subs (B); B=sym2poly (B); A=subs (A); A=sym2poly (A);
  f=f0 * logspace (-2, 2, 101); H=freqs (B, A, f * 2 * pi);
  subplot (211); semilogx (f, abs (H),'b');
  tstr= ['alfa = ' num2str (alfa) ', R = ' num2str (R) …
    ' Ohm, C = ', num2str (C) ' F'];
  title (tstr);      ylabel ('abs (H)'); grid;
  subplot (212); semilogx (f, angle (H) * 180/pi,'b'); grid;
  xlabel ('frequency (Hz)'); ylabsl ('angle (H) o');
  end;   set (gcf,'Name','Double-T Network','Num','off');
end;
% ==================================
function B =s_poly (A);                       % 系数化简函数
syms s alfa;
B=collect (A);
b3-diff (B,'s', 3) /6;                        b3=simple (b3);
b2=diff (B-b3 * s^3,'s', 2) /2;               b2=simple (b2);
```

```
b1 = diff (B - b3 * s^3 - b2 * s^2,'s');          b1 = simple (b1);
b0 = B - b3 * s^3 - b2 * s^2 - b1 * s;            b0 = simple (b0);
B = b3 * s^3 + b2 * s^2 + b1 * s + b0;
```

双 T 选频网络的转移函数和输入阻抗表达式的符号运算结果见前面，幅频特性曲线见图 5.5。

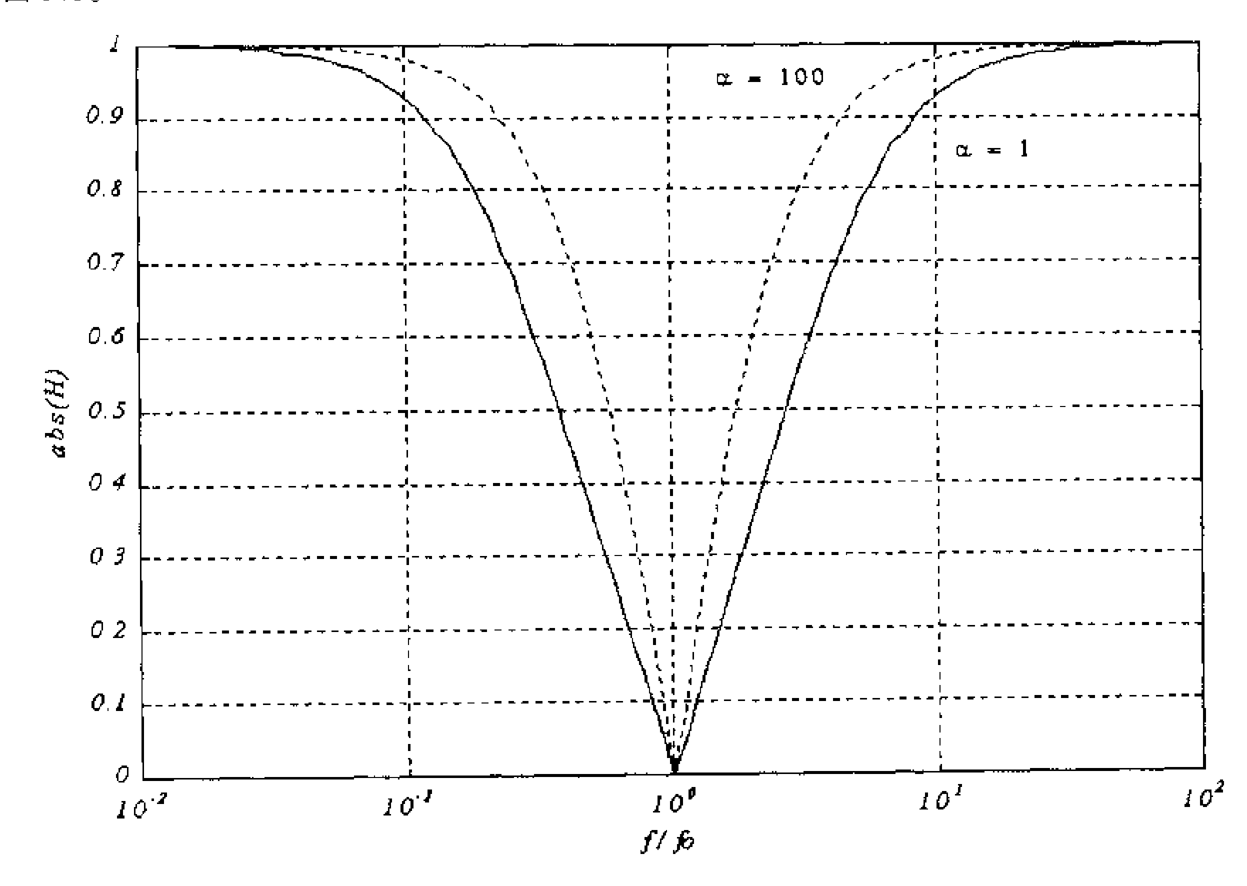


图 5.5 双 T 选频网络的幅频特性曲线

5.1.4 模拟滤波器

模拟滤波器的设计与无限冲击响应滤波器的设计是使用相同的 M 函数，通常可以采用巴特伍兹滤波器、切比雪夫滤波器等形式。

例 2 二阶阻容网络

由电阻和电容组成的二阶网络是经常使用的，其结构主要有三种方式：低通、高通、带通，见图 5.6。分析二阶网络函数的转移函数和输入阻抗等问题，使用符号运算方式是很方便的。根据 A 矩阵相乘的方式，就可以求出二阶网络的全部参数。

1) 低通网络的转移函数

```
》syms R1 R2 C1 C2 s;
》A= [1 R1; 0 1] * [1 0; s*C1 1] * [1 R2; 0 1] * [1 0; s*C2 1];
```

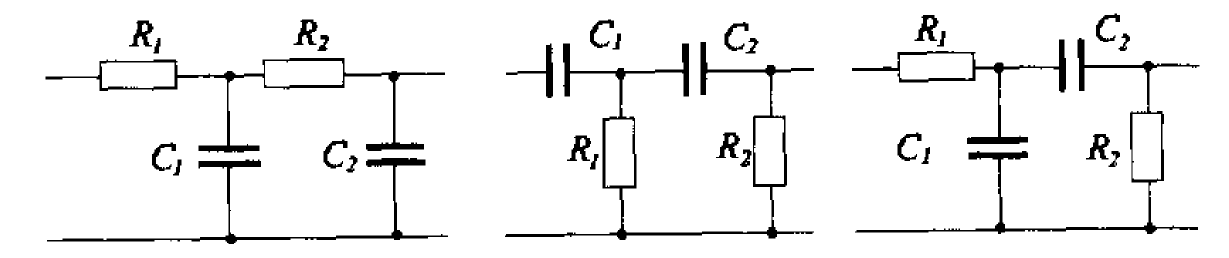
```
》H=collect (1/A (1, 1)); pretty (H)
                              1
           ----------------------------------------
                         2
           R1 C1 R2 C2 s  + (R1 C1 + (R2+R1) C2) s + 1
```

2）高通网络的转移函数

```
》syms R1 R2 C1 C2 s
》A= [1 1/s/C1; 0 1] * [1 0; 1/R1 1] * [1 1/s/C2; 0 1] * [1 0; 1/R2 1];
》H=1/A (1, 1); [B, A] =numden (H); B=collect (B); A=collect (A);
》pretty (B/A)
                              2
                       R1 s C1 C2 R2
          ------------------------------------------
                  2
          R1 s  C1 C2 R2 + (R1 C2 + C2 R2 + R1 C1) s + 1
```

3）带通网络的转移函数

```
》syms R1 R2 C1 C2 s
》A= [1 R1; 0 1] * [1 0; s*C1 1] * [1 1/s/C2; 0 1] * [1 0; 1/R2 1];
》H=1/A (1, 1); [B, A] =numden (H); B=collect (B); A=collect (A);
》pretty (B/A)
                         s C2 R2
          ------------------------------------------
                 2
          R1 s  C1 C2 R2 + (R1 C2 + C2 R2 + R1 C1) s + 1
```



(a) 低通二阶网络　　(b) 高通二阶网络　　(c) 带通二阶网络

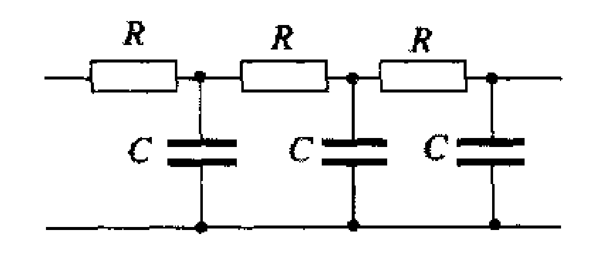
图 5.6　电阻电容组成的二阶网络

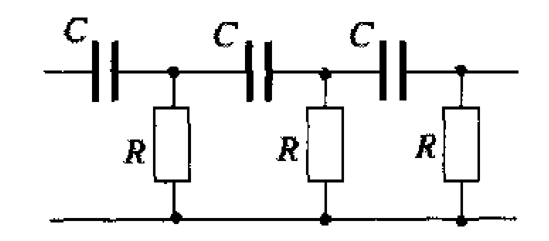
例 3　三阶阻容网络

图 5.7 中给出了低通型和高通型的两种三阶阻容网络，使用通常的手段分析它们的转移函数是十分繁杂的工作，而使用 MATLAB 进行仿真则是十分方便的，其方法也是将每个元件的 A 矩阵相乘，从而得到网络总的 A 矩阵，进一步就可得到转移函数、输入阻抗等参数。

1）低通网络的转移函数

```
》syms R C s;
》A= [1 R; 0 1] * [1 0; s*C 1] * [1 R; 0 1] * [1 0; s*C 1] * [1 R; 0 1]
* [1 0; s*C 1];
》H=collect (1/A (1, 1)); pretty (H)
                       1
          -----------------------------
           3 3 3        2 2 2
          R C s  + 5 R C s  + 6 R C s + 1
```





(a) 低通三阶网络　　(b) 高通三阶网络

图 5.7 电阻电容组成的三阶网络

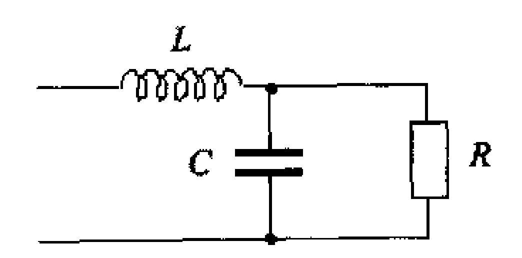
2) 高通网络的转移函数

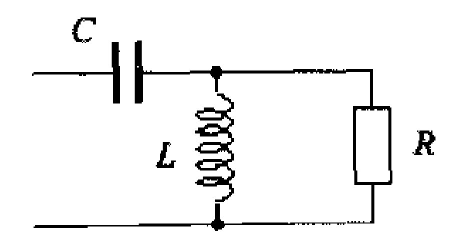
```
》syms R C s
》A= [1 1/s/C; 0 1] * [1 0; 1/R 1] * [1 1/s/C; 0 1] * [1 0; 1/R 1];
》A=A* [1 1/s/C; 0 1] * [1 0; 1/R 1];
》H=1/A (1, 1); [B, A] =numden (H); B=collect (B); A=collect (A);
》pretty (B/A)
```

$$\frac{R^3C^3s^3}{R^3C^3s^3 + 6R^2C^2s^2 + 5RCs + 1}$$

例 4　模拟二阶巴特伍兹滤波器

模拟二阶巴特伍兹滤波器可以由图 5.8 中的电路来实现。先使用符号运算法计算出网络的转移函数，接着用 butter 函数计算出巴特伍兹滤波器的系数向量，然后通过系数对比的方法来确定电阻和电容的数值。





(a) 低通滤波器　　(b) 高通滤波器

图 5.8 二阶巴特伍兹滤波器

1) 低通滤波器

计算转移函数

```
》syms L C R s
》A= [1 s*L; 0 1] * [1 0; s*C 1] * [1 0; 1/R 1];
》H=1/A (1, 1); pretty (collect (H))
```

$$\frac{1}{1 + s^2 LC + \frac{sL}{R}}$$

设计一个截止频率为 2000Hz 的二阶低通巴特伍兹滤波器，

```
》[b, a] =butter (2, 2000*2*pi,'s'); b=b/a (3); a=a/a (3);
```
计算结果为：b：[　0　0　1.0000e+000]

a：[6.3326e-009　1.1254e-004　1.0000e+000]

显然 LC=a (1)，L/R=a (2)，设 R = 600Ω，于是

```
》R=600; L=a (2) *R, C=a (1) /L,
L = 6.7524e-002      C = 9.3783e-008
```

为了对滤波器的设计进行验证，可将 L=67.524mH、C=0.093783μF、R=600Ω 等元件参数代入 A 矩阵，然后计算它的幅频响应特性，其结果与预期的数值完全一样。

```
》syms s;
》C=9.3783e-08; L=6.7524e-02; R=600;
》A= [1 s*L; 0 1] * [1 0; s*C 1] * [1 0; 1/R 1];
》H=1/A (1, 1); [B, A] =numden (H); B=sym2poly (B); A=sym2poly (A);
》f=20: 20: 20000; H=freqs (B, A, 2*pi*f);
》semilogx (f, 20*log10 (abs (H))); grid; zoom xon; hold on;
》plot ( [2000 2000], [0 -10],'k:'); hold off;
```

2）高通滤波器

计算转移函数

```
》syms L C R s
》A= [1 1/s/C; 0 1] * [1 0; 1/s/L 1] * [1 0; 1/R 1];
》H=1/A (1, 1); [B, A] =numden (H); B=collect (B); A=collect (A);
》pretty (B/A)

                 2
                s  C L R
            ---------------
             2
            s C L R + R + s L
```

设计一个截止频率为 2000Hz 的二阶高通巴特伍兹滤波器，

```
》[b, a] =butter (2, 2000*2*pi,'high','s'); b=b/a (3); a=a/a (3);
```
计算结果为：b：　[6.3326e-009　0　0]

a：　[6.3326e-009　1.1254e-004　1.0000e+000]

显然，电阻、电容和电感的数值与低通滤波器是完全一样的。

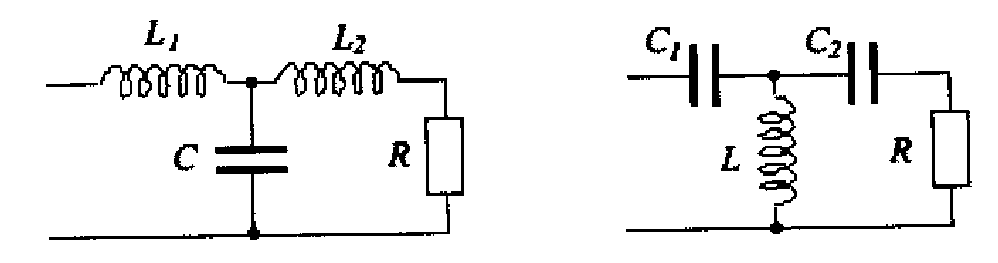


图 5.9　三阶滤波器

例 5　模拟三阶切比雪夫滤波器

模拟三阶滤波器可以由图 5.9 中的电路来实现，而切比雪夫滤波器和巴特伍兹滤波器的

区别是两者的参数不同。

1）低通滤波器

计算转移函数

```
》syms Ll L2 C R s
》A= [1 s*L1; 0 1] * [1 0; s*C 1] * [1 s*L2; 0 1] * [1 0; 1/R 1];
》H=1/A (1, 1); pretty (collect (H))
```

$$\frac{1}{\frac{L1\,C\,L2\,s^3}{R}+s^2 L1\,C+\frac{(L2+L1)\,s}{R}+1}$$

设计一个截止频率为 2000Hz 的三阶低通切比雪夫滤波器，波纹为 1dB。

```
》[b, a] =cheby1 (3, 1, 2000*2*pi,'s'); b=b/a (4); a=a/a (4);
```

计算结果为：

```
b: [0              0              0              1.0000e+000]
a: [1.0257e-012    1.2739e-008    2.0059e-004    1.0000e+000]
```

显然 L2/R=a(1)/a(2),(L2+L1)/R=a(3), L1*C=a(2), 设 R = 600Ω,于是

```
》R=600; L2=a (1) /a (2) *R, L1=a (3) *R-L2, C=a (2) /L1,
L2= 4.8310e-002, L1 = 7.2042e-002, C = 1.7683e-007
```

将 $L_1=72.042$mH、$L_2=48.31$mH、$C=0.17683\mu$F、$R=600\Omega$ 等元件参数代入 A 矩阵，使用下面的程序段计算它的幅频响应特性，其结果见图 5.10。

```
》syms s;
》C=1.7683e-007; L1=7.2042e-002; L=4.8310e-002; R=600;
》A= [1 s*L1; 0 1] * [1 0; s*C 1] * [1 s*L2; 0 1] * [1 0; 1/R 1];
》H=1/A (1, 1); [B, A] =numden (H); B=sym2poly (B); A=sym2poly (A);
》f=20: 20: 20000; H=freqs (B, A, 2*pi*f);
》semilogx (f, 20*log10 (abs (H))); grid; zoom on; hold on;
》plot ( [2000 2000], [0 -10],'k:'); hold off;
```

2）高通滤波器

计算转移函数

```
》syms L C1 C2 R s
》A= [1 1/s/C1; 0 1] * [1 0; 1/s/L 1] * [1 1/s/C2; 0 1] * [1 0; 1/R 1];
》H=1/A (1, 1); [B, A] =numden (H); B=collect (B); A=collect (A);
》pretty (B/A)
```

$$\frac{s^3\,C1\,L\,C2\,R}{s^3\,C1\,L\,C2\,R+(L\,C2+C1\,L)\,s^2+s\,C2\,R+1}$$

设计一个截止频率为 2000Hz 的三阶高通切比雪夫滤波器，波纹为 1dB。

```
》[b, a] =cheby1 (3, 1, 2000*2*pi,'high','s'); b=b/a (4); a=a/a (4);
b: [2.4758e-013              0              0              0]
```

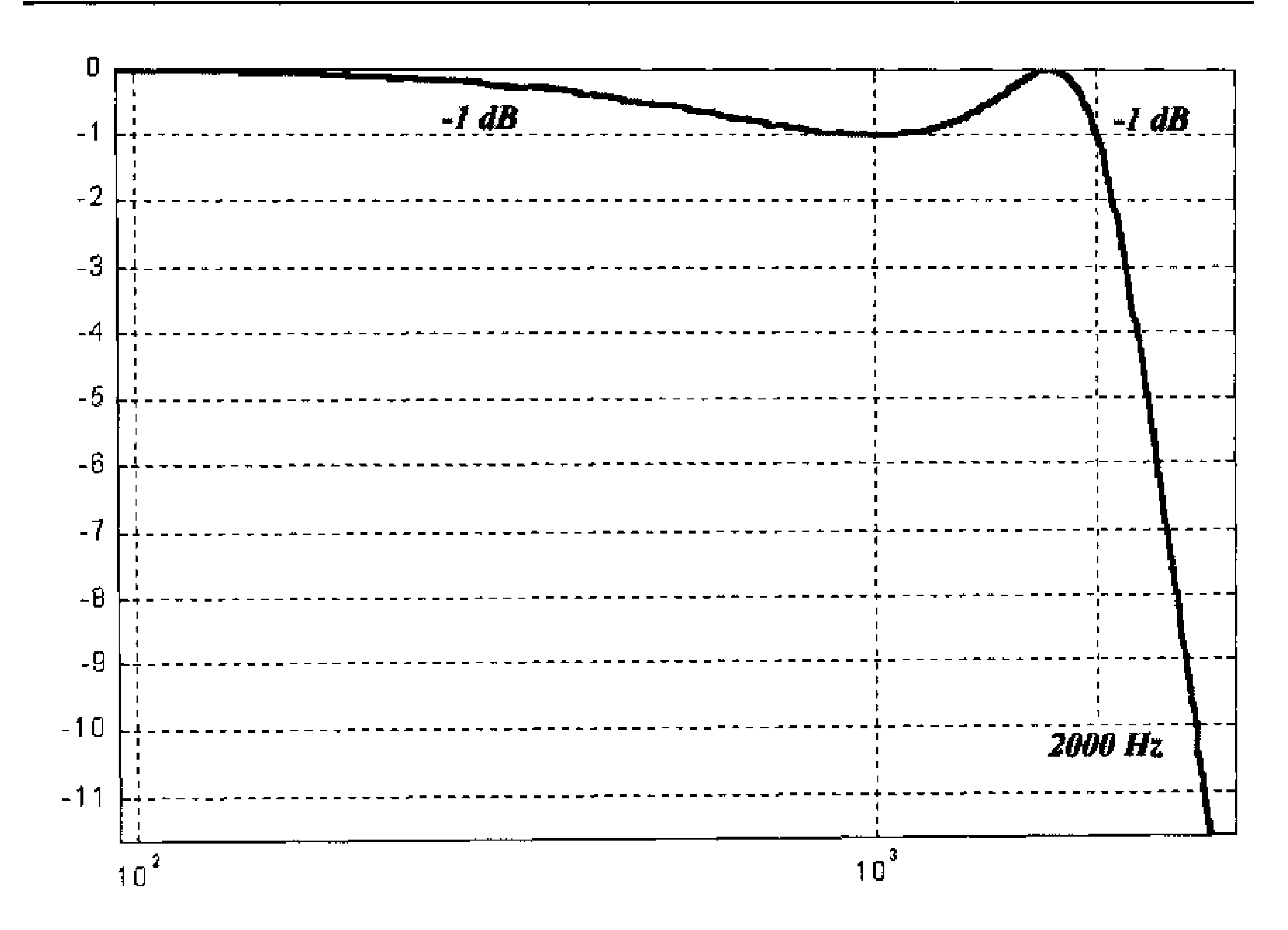


图 5.10 三阶切比雪夫低通滤波器的幅频特性

a: [2.4758e-013 7.8423e-009 7.8650e-005 1.0000e+000]

对比系数有：C2 R=a (3), C1 L C2 R=a (1), L (C1+C2) =a (2), 设 R = 600Ω, 于是

》R=600; C2=a (3) /R, L= (a (2) -a (1) /a (3)) /C2, C1=a (1) /a (3) /L,

C2 = 1.3108e-007, L = 3.5812e-002, C1 = 8.7901e-008

经验算，幅频特性与预期的结果完全一样。

5.1.5 音调控制

音频放大器的音调控制电路也可以看作是一种两端口网络，自定义的 M 函数 tone.m 用来对常见的电压型衰减式音调控制电路进行仿真，这种音调控制电路的电路图见图 5.11。电压型衰减式音调控制电路，最早见于 1949 年 6 月号的《音频工程》杂志上，它在各种音频设备中使用得相当广泛，其控制范围在高频端和低频端分别到达 ± 15dB 以上，其结构也不算复杂，同时工作稳定，又不产生非线性失真，因此它受到众多音频设备生产厂家和业余无线电爱好者的青睐。VR1 为高频控制电位器，而 VR2 为低频控制电位器，为了均匀地进行控制，两个电位器应该采用指数型的，当滑臂位于中间位置时，滑臂与地之间的电阻数值应为总阻值的 9.09%。

电压型衰减式音调控制电路可以看作是两个并联的两端口网络，一个两端口网络是高频控制电位器和两个电容组成，其余的 6 个元件构成另外一个两端口网络。先使用符号运算方

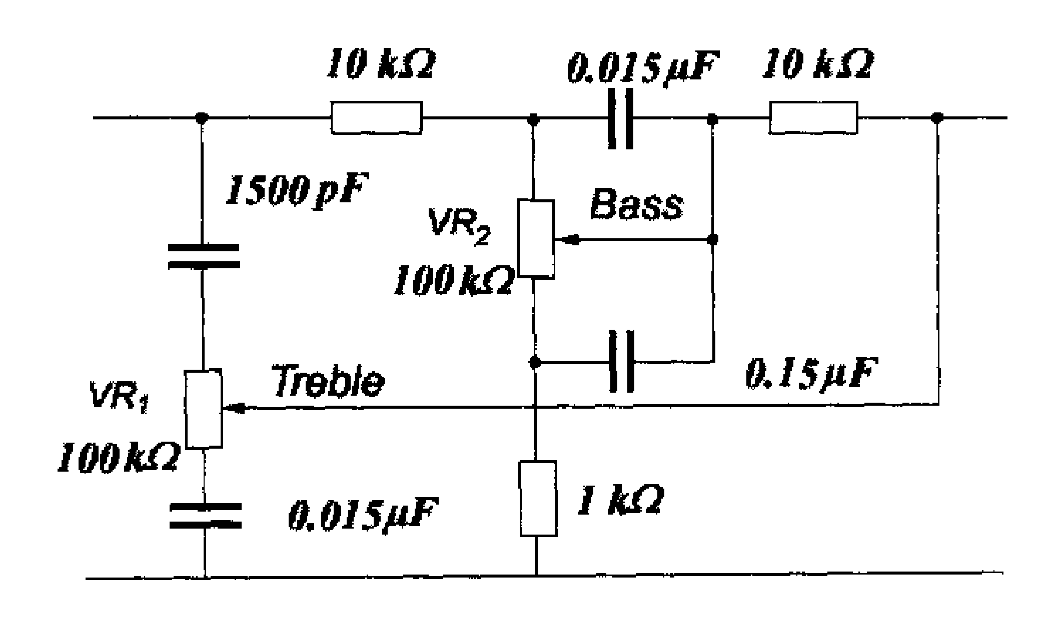


图 5.11 电压型衰减式音调控制电路

式来求出该网络的转移函数，然后再根据两个电位器的参数来绘制整个音调控制电路的幅频特性曲线。tone 函数应在 5.x 版的 MATALB 环境下使用，该函数采用了按钮式界面，其中使用了两个滑标来模拟音调控制电位器，另外三个按钮分别代表着高低音同时提升、高低音同时衰减和平坦三种状态。程序清单如下：

```
function [ ] = tone (action);                % 建立按钮式界面，调用相应的函数
%
%  Simulation of Tone Control,   BBI 2000
global Kt Kb B A H1 H2 hr;          % 定义全程变量
if nargin<1;
  action='initialized';
end;
if strcmp (action,'initialized');      % 建立按钮式界面
  Kt=.5; Kb=.5; Km=0; hr= [];
  figure ('Name','TONE CONTROL, BBI 2000','Num','off', …
    'Units','Normal','Pos', [0.2 0.2 .7 .7],'Color', [1 1 1]);
  axes ('Units','Normal','Pos', [0.08 0.1 .7 .84],'vis','off');
  uicontrol ('Style','Frame','Units','Normal', …
    'Pos', [.82 0 .2 1],'Back', [.2 .4 .4]);
  uicontrol ('Style','Text','Units','Normal','Pos', …
     [.84 .75 .15 .04],'String','TREBLE','Back', [.8 .8 .8]);
  H1=uicontrol ('Style','Slider','Units','Normal', …
    'Pos', [.84 .7 .15 .04],'Back', [.8 .8 .8],'Value', Kt, …
    'Callback','tone (''Treble'');');
```

```
  uicontrol ('Style','Text','Units','Normal', …
    'Pos', [.84 .55 .15 .04],'String','BASS','Back', [.8 .8 .8]);
  H2=uicontrol ('Style','Slider','Units','Normal',
    'Pos', [.84 .5 .15 .04],'Back', [.8 .8 .8],'Value', Kb, …
    'Callback','tone (''Bass'');');
  uicontrol ('Style','Push','Units','Normal', …
    'Pos', [.84 .4 .15 .05],'String','INCREASE', …
    'Callback','tone (''Increase'');');
  uicontrol ('Style','Push','Units','Normal', …
    'Pos', [.84 .3 .15 .05],'String','DECREASE', …
    'Callback','tone (''Decrease'');');
  uicontrol ('Style','Push','Units','Normal', …
    'Pos', [.84 .2 .15 .05],'String','RESTORE', …
    'Callback','tone (''Restore'');');
  [B, A] =tone2; tone1 (.5, .5);
elseif strcmp (action,'Treble');                    % 高音调节
  Kt=get (gco,'Value');   Kt=round (Kt*200) /200; tone1 (Kt, Kb);
elseif strcmp (action,'Bass');                      % 低音调节
  Kb=get (gco,'Value');   Kb=round (Kh*200) /200; tone1 (Kt, Kb);
elseif strcmp (action,'Restore');                   %平坦特性
  Kb=.5;   Kt=.5;     tone1 (Kt, Kb, 1);
  set (H1,'value', .5);     set (H2,'value', .5);
elseif strcmp (action,'Increase');                  % 高音低音同时提升
  h=tone1 (1, 1);      hr= [hr; h];
  set (H1,'value', 1);   set (H2,'value', 1);
elseif strcmp (action,'Decrease');                  % 高音低音同时衰减
  h=tone1 (0, 0);      hr= [hr; h];
  set (H1,'value', 0);   set (H2,'value', 0);
end;
% ================================
function [h] =tone1 (Kt, Kb, Km);          % 辅助函数，用来计算幅频特性
if nargin<3; Km=0; end;
global hr A B;   syms s;
f=20: 20: 15000;    [m, n] =size (hr);    x=3.4594;
if Km= =0;
  b=sym2poly (subs (B)); a=sym2poly (subs (A)); h=freqs (b, a, 2*pi*f);
elseif Km= =1;
  h=ones (size (f)) *10^ (-20.828/20);
end;
```

```
if m>0;
  Hm=semilogx (f, 20*log10 (abs (hr (1,:))) +20.828);
  set (Hm,'color', [1 1 1] *.8);      hold on;
  if m>1;
    Hm=semilogx (f, 20*log10 (abs (hr (2,:))) +20.828);
    set (Hm,'color', [1 1 1] *.8);
  end;
end;
semilogx (f, 20*log10 (abs (h)) +20.828);    hold off;    grid;
axis ( [20 20000 -20 +20]);    xlabel ('frequency (Hz)');
bstr=num2str (Kb*200-100);    tstr=num2str (Kt*200-100);
title ( ['Bass ' bstr '% , Treble ' tstr '%']);
% =====================================
function [b, a] =tone2 ()              % 符号法计算转移函数
syms s Kt Kb x;
C1=1500e-12;        C2=0.015e-6;        C3=0.015e-06;        C4=0.15e-06;
R3=10e+03;          R4=1.0e+03;         R5=10e+03;
VR1=100e+03;        VR2=100e+03;                             % 确定元件的参数
A1= [1 VR1* (1-Kt^x) +1/s/C1; 0 1] * [1 0; 1/ (VR1*Kt^x+1/s/C2) 1];
Z1= [A1 (1, 1) det (A1); 1 A1 (2, 2)] /A1 (2, 1);           % 高音控制电路
Z3=1/ (s*C3+1/ (VR2* (1-Kb^x))); Z4=1/(s*C4+1/(VR2*Kb^x))+R4;
A2= [1 R3+Z3; 0 1] * [1 0; 1/Z4 1] * [1 R5; 0 1];
Z2= [A2 (1, 1) det (A2); 1 A2 (2, 2)] /A2 (2, 1);           % 低音控制电路
Y=inv (Z1) +inv (Z2); Z=inv (Y);                            % 网络并联
A= [Z (1, 1) det (Z); 1 Z (2, 2)] /Z (2, 1);
[a, b] =numden (A (1, 1));
a=collect (a,'s');     b=collect (b,'s');         % 转移函数的分子与分母
```

从仿真的结果可以看出，这个电压型衰减式音调控制电路是很有效的，其低音（30Hz）的控制范围是 +19dB、-18dB，高音（10000Hz）的控制范围是 +17dB、-14 dB，当两个电位器均位于中间位置时，幅频特性是完全平坦的。图中给出的结果是：低音提升 6%，高音提升 4%。另外，根据计算此音调控制电路本身对信号产生约 20dB 的接入损耗，因此其后必须安排一级放大，以补偿这个损耗。

在音调控制电路的设计工作中，使用系统仿真的方法可以用来进行模拟试验，以便得到比较好的性能。

5.2 预加重与去加重网络

在调频广播系统中采用了预加重和去加重来提高系统的信噪比，由于白噪声在通过调频解调器之后，噪声功率主要集中在了高频端，形成了所谓的“三角噪声”。在发射端采用预

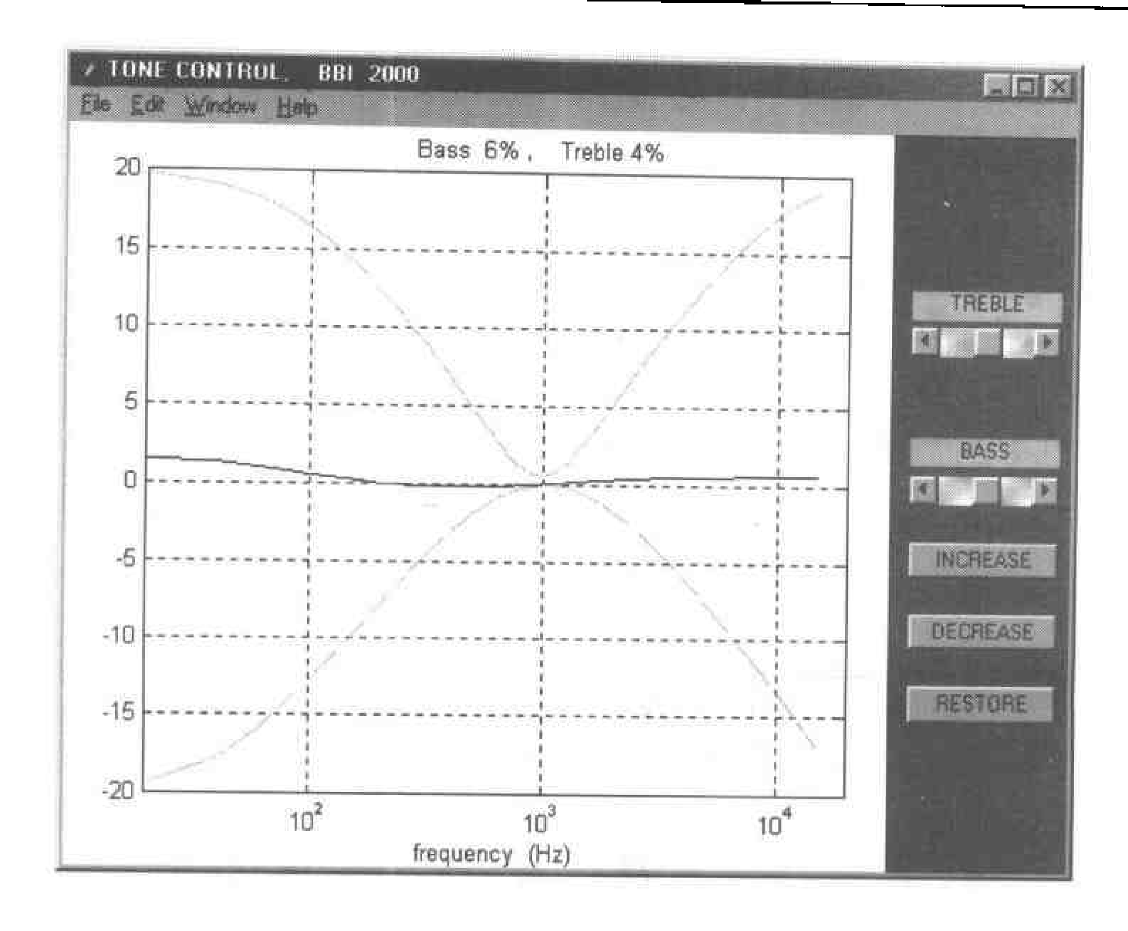


图 5.12　音调控制电路仿真

加重网络，提高高频端的增益，而在接收端则采用去加重网络，降低高频端的增益。这样对于信号来说，整个系统的幅频特性是平坦的，接收机内的去加重网络有效地衰减了高频噪声，因此提高了系统的信噪比。

此处对卫星电视广播系统中使用的视频去加重网络和音频去加重网络进行仿真，去加重网络由桥 T 形网络组成，见图 5.13。在国际无线电咨询委员会（现为 ITU－R）的 405－1 建议书中对卫星广播中的视频去加重网络进行了规定，定义了两种曲线，一种用于 625 行系统，另一种用于 525 行系统。在国际电话电报咨询委员会（现在为 ITU－T）的 J.17 建议中对卫星广播中的音频去加重网络进行了规定，通常称为 J17 曲线。同时在某些卫星广播系统中还使用着比较简单的去加重网络，该网络由一个电阻和一个电容组成，其时间常数为 75μs。

视频去加重网络和音频去加重网络的结构是一样的，都是对称的桥 T 形网络，负载阻抗为 R，两个网络的差别在于元件的参数不同，显然去加重网络为二阶网络，由于网络的结构比较复杂，因此采用通常的方法来求其转移函数是十分繁杂的。分析桥 T 形网络时，可以采用 Π－T 变换的方式将上部的 Π 型网络变换成 T 形网络，这样桥 T 形网络就变成了 T 形网络，然后使用符号运算的方式求出其转移函数，进一步就得到了它的幅频特性。自定义的 **deemphasis** 函数就是用来分析去加重网络的，它只能在 5.x 版下运行，其用法是：

› deemphasis;　　　　　给出 CCIR 405－1 建议书中规定的两种视频去加重曲线。

› deemphasis ('a');　　给出 J.17 建议中规定的音频去加重曲线和时间常数分别为 75μs

和 50μs 的音频去加重曲线。

采用符号运算方式得到的转移函数经过适当的整理后，其表达式为：

$$H(s)=\frac{b_1s^2+b_2s+2RR_1R_2}{a_1s^2+a_2s+2RR_1R_2}$$

其中，

$$b_1=[R_2^2+R_3(R_1+2R_2)]RLC$$
$$b_2=[R_1R_2C(R_2+2R_3)+L(R_1+2R_2)]R$$
$$a_1=[R_2(R_2+2R_3)(R+R_1)+RR_1(R_2+R_3)]LC$$
$$a_2=L[RR_1+2R_2(R+R_1)]+RR_1R_2C(R+2R_3)$$

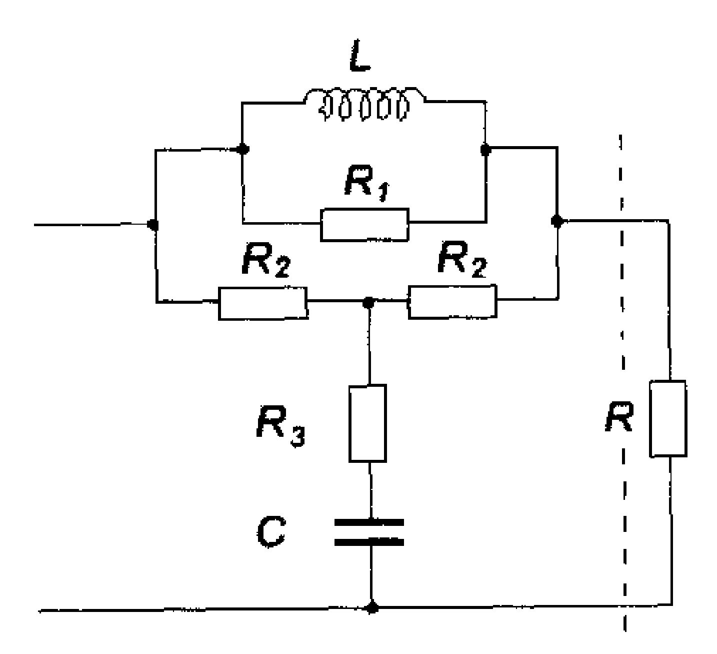


图 5.13 桥 T 形去加重网络

函数 deemphasis.m 的清单为：

```
function [] =deemphasis (Kstr);
%
%  BBI  1999
if nargin<1; Kstr='v'; end;
syms R1 R2 R3 R L C s
Z11=s*R1*L/ (s*L+R1);                          %pi-tee 变换
Z1=Z11*R2/ (Z11+2*R2);     Z2=Z1;
Z3=R2*R2/ (Z11+2*R2);
A= [1 Z1; 0 1] * [1 0; 1/ (Z3+R3+1/s/C) 1] * [1 Z2; 0 1] * [1 0; 1/R 1];
H=1/A (1, 1); H=simplify (H);        [B, A] =numden (H);
Bsym=collect (B);    Asym=collect (A);          %转移函数的分子和分母
```

```
if strcmp (Kstr,'v');                                    %视频去加重
  R1=300; R2=75; R3=18.75; R=75; L=30.53e-06; C=5424e-12;
  B=subs (Bsym);    B=sym2poly (B);    A=sym2poly (subs (Asym));
  f=.01: .01: 10; f=f*1e+06;       H=freqs (B, A, 2*pi*f);
  semilogx (f, 20*log10 (abs (H)),'b'); hold on;
  R1=275.8; R2=75; R3=20.4; R=75; L=50.16e-06; C=8917e-12;
  B=subs (Bsym); B=sym2poly (B);
  A=sym2poly (subs (Asym));
  f=.01: .01: 10;    f=f*1e+06; H=freqs (B, A, 2*pi*f);
  semilogx (f, 20*log10 (abs (H)),'r');
  nameStr='VIDEO DEEMPHASIS CURVE, CCIR 405-1';
  text (2.3e+05, -4.8,'A','FontName','Arial','FontSize', 10);
  text (5.2e+05, -4.5,'B','FontName','Arial','FontSize', 10);
elseif strcmp (Kstr,'a');                                %音频去加重
  R1=4596; R2=600; R3=78.3; R=600; L=176.9e-03; C=0.491e-6;
  B=subs (Bsym);      B=sym2poly (B);
  A=sym2poly (subs (Asym));
  f=10: 10: 20000;       H=freqs (B, A, 2*pi*f);
  semilogx (f, 20*log10 (abs (H)),'r'); hold on;
  B= [0 1]; A= [75e-06 1]; H=freqs (B, A, 2*pi*f);
  semilogx (f, 20*log10 (abs (H)),'b');
  B= [0 1]; A= [50e-06 1]; H=freqs (B, A, 2*pi*f);
  semilogx (f, 20*log10 (abs (H)),'m');
  nameStr='AUDIO DEEMPHASIS CURVE';
  text (120, -3,'J.17','FontName','Arial','FontSize', 10);
  text (1100, -3,'75 μs','FontName','Arial','FontSize', 10);
  text (3500, -3,'50 μs','FontName','Arial','FontSize', 10);
end;
grid on; zoom xon; hold off;
xlabel ('Frequency (Hz)'); ylabel ('Attenuation (dB)');
set (gcf,'Name', nameStr,'NumberTitle','off');
```

图 5.14　给出了音频去加重特性的计算曲线。

5.3　电缆均衡器

均衡器是电缆电视系统中的一个重要部件，它用来补偿电缆产生的幅频失真。在电缆电视系统中使用较长距离的电缆将信号从系统的前端或分前端传送至远处的分配系统中，这样的电缆称为干线电缆。每一段干线电缆的长度通常在数百米左右，为了弥补此段电缆的损耗，在后面要加上干线放大器。电缆产生的衰减大约与频率的平方根成正比，电缆在低频端

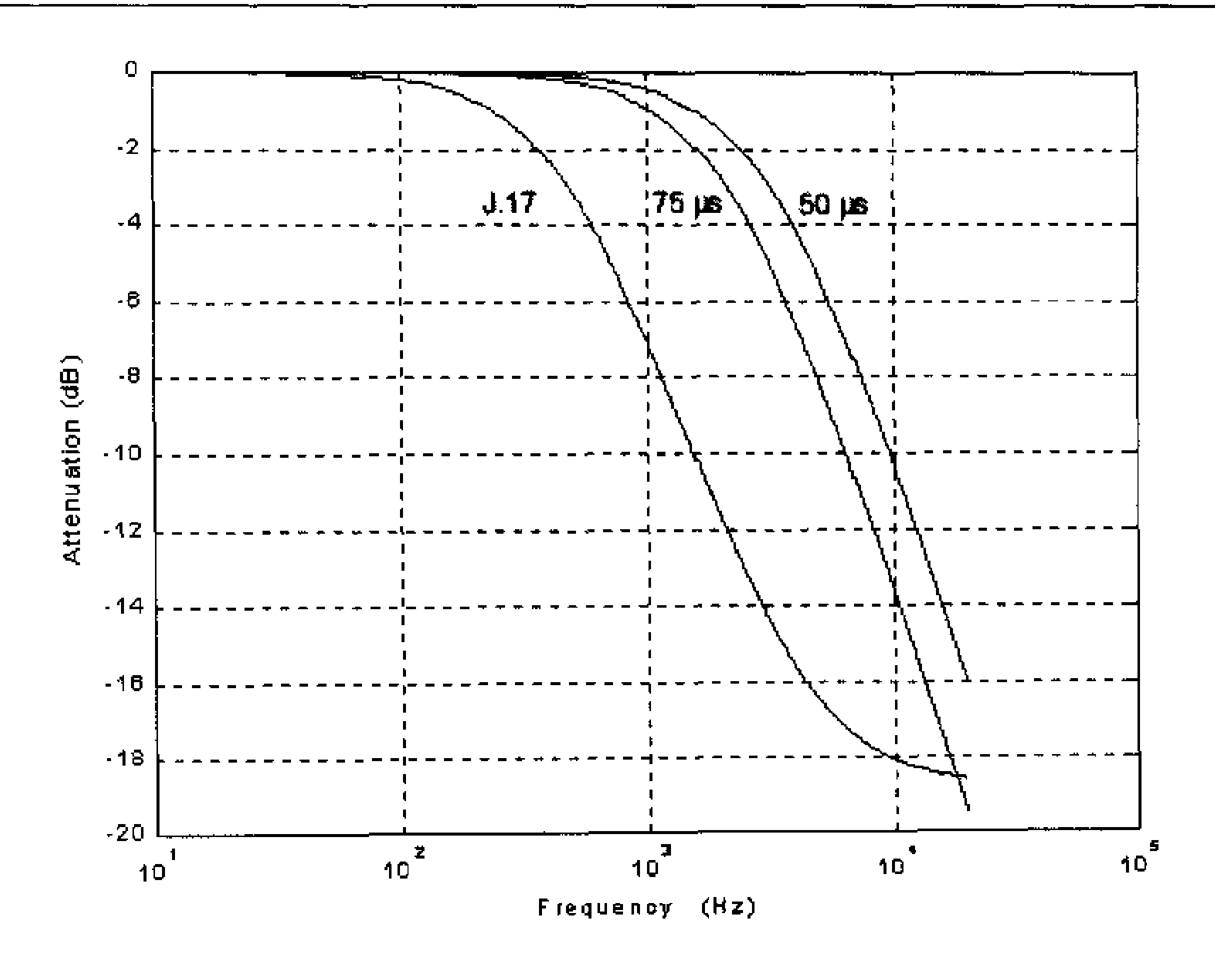


图 5.14 卫星广播系统中的音频去加重曲线

的衰减相对比较小，而在高频端的衰减则比较大，因此在干线放大器处要使用电缆均衡器，其幅频特性正好与电缆相反，频率越高衰减越小，频率越低衰减反而越大，这样就可以补偿电缆产生的幅频失真。均衡器有不同的规格，分别用来补偿不同长度、不同型号的干线电缆。

电缆均衡器的典型结构如图 5.15 所示，它也是一个桥 T 形网络，其中有一个串联谐振回路和一个并联谐振回路，两者的谐振频率相同，虚线外面的是负载电阻。显然当均衡器谐振时，均衡器的衰减是最小的，而当频率很低时，电容容抗变得很大，而电感感抗变得很小，所以此时的均衡器就相当于电阻组成的桥 T 形衰减器。

分析电缆均衡器也可以采用 Π－T 变换的方式，然后求出它的转移函数。当频率趋于 0 时均衡器相当于一个衰减器，其转移函数和输入阻抗可由下面的程序段求出：

```
》syms R0 R1 R2
》z=R0+R1+R0; z1=R0*R1/z; z2=R0*R0/z; z3=R0*R1/z; R2=R0*R0/R1;
》A=[1 z1; 0 1]*[1 0; 1/(R2+z2) 1]*[1 z3; 0 1]*[1 0; 1/R0 1];
》A=simple(A); H=1/A(1, 1), Zin=A(1, 1)/A(2, 1)
```

运算结果为：H=R0/(R0+R1)　　Zin = R0

将上述转移函数的数值换算成为 dB，这就是设计均衡器的主要依据。

自定义的 equilizer.m 函数用来分析电缆均衡器，其用法是：

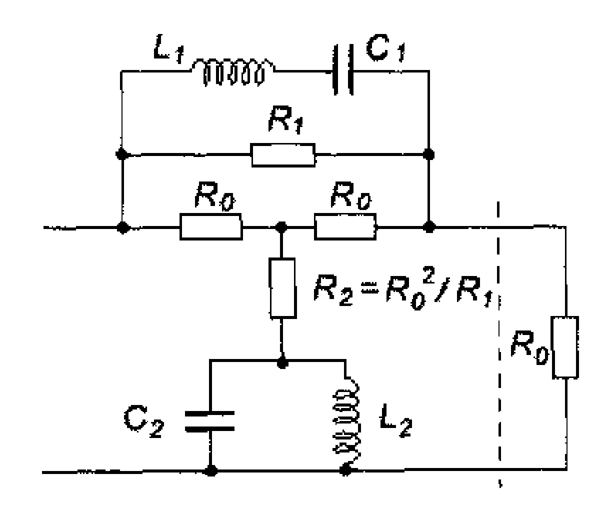


图 5.15　电缆均衡器

》[L1, L2, C1, C2, R1, R2] =equilizer (C1, C2, f0, att)　　C1 和 C2 的单位为 pF；f0 为电缆系统的上限工作频率，单位为 MHz；att 为均衡器的均衡量，单位为 dB。

程序清单如下：

```
function [L1, L2, C1, C2, R1, R2] =equilizer (C1, C2, f0, att);
%
%   BBI 2000
if nargin<4; att=12; end;            % dB
if nargin<3; f0=450; end;            % MHz
if nargin<2; C2=50; end;             % pF
if nargin<1; C1=6;    end;           % pF
syms s;
att=att+3;          R0=75;
R1= (10^abs (att/20) -1) * R0;       R2=R0^2/R1;
C1=C1 * 1e-12;       C2=C2 * 1e-12;
f= (50: f0) * 1e+06; f0=f0 * 2e+06;    w0=2 * pi * f0;
L1=1/ (w0^2 * C1);     L2=1/ (w0^2 * C2);
zz1=1/ (1/R1+1/ (s* L1+1/s/C1));
z=R0+R1+R0; z1=R0 * zz1/z; z2=R0 * R0/z; z3=R0 * zz1/z;
A= [1 z1; 0 1] * [1 0; 1/ (R2+1/ (s* C2+1/s/L2) +z2) 1];
A=A* [1 z3; 0 1] * [1 0; 1/R0 1];
[a, b] =numden (A (1, 1)); a=sym2poly (a); b=sym2poly (b);
H=freqs (b, a, 2 * pi * f); H=20 * log10 (abs (H)); a0=max (H);
semilogx (f * 1e-06, H-a0);    grid;
xlabel ('Frequency (MHz)'); ylabel ('Ralative Attenuation (dB)');
title ( ['Attenuation ' num2str (round ( -a0 * 10) * .1) ' (dB)']);
```

```
text (12, -.6, ['L1 = ' num2str (L1 * 1e+06) ' μH']);
text (12, -1.6, ['L2 = ' num2str (L2 * 1e+06) ' μH']);
text (12, -2.6, ['C1 = ' num2str (C1 * 1e+12) ' pF']);
text (12, -3.6, ['C2 = ' num2str (C2 * 1e+12) ' pF']);
text (300, -2.6, ['R1 = ' num2str (R1) ' ohm']);
text (300, -3.6, ['R2 = ' num2str (R2) ' ohm']);
text (40, -att+3.5, num2str (50));
f0=f0 * .5e-06;    text (f0, -.5, num2str (f0));
```

使用 equilizer 函数可以对均衡器进行仿真，从而对均衡器进行优化设计。图 5.16 为一个 450MHz 系统的 12dB 均衡器的特性曲线和参数。

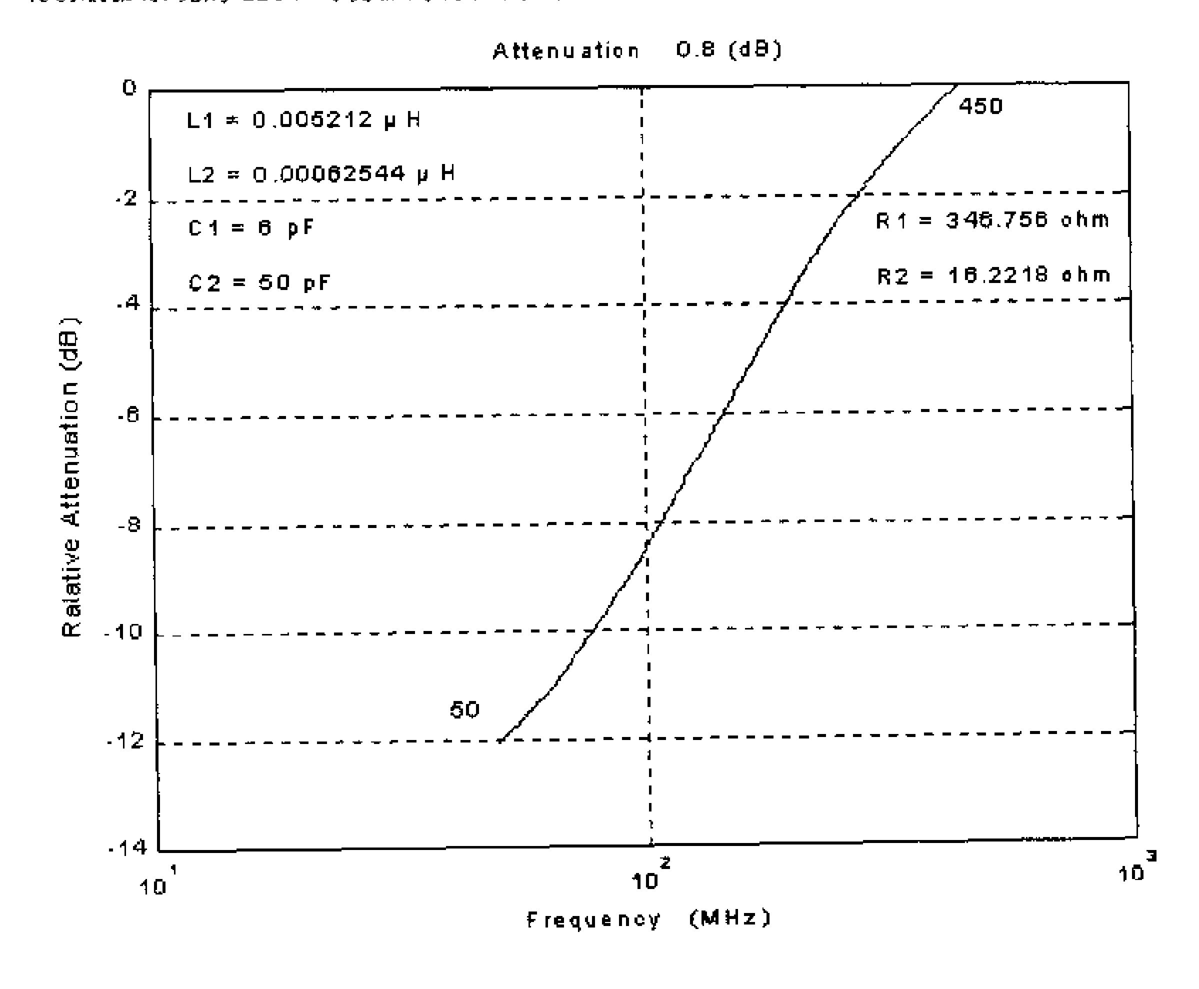


图 5.16 12dB 均衡器（450MHz 系统）

5.4 双调谐回路

谐振电路的应用是很广泛的。谐振电路由电感和电容组成，有串联谐振电路、并联谐振电路和双调谐回路（亦称为耦合谐振回路）等形式。通常，分析它的谐振频率和频率特性是很繁琐的工作，因此使用 MATLAB 对它进行仿真是很必要的。

两个单谐振回路彼此之间按一定的方式进行耦合，就构成了双调谐回路。利用双调谐回路可以获得比单调谐回路更好的谐振特性，同时进行阻抗变换也比较方便。图 5.17a 给出了一个典型的互感耦合双调谐回路，实际上它是一个晶体管谐振放大器的物理模型，信号源为电流源。为了便于进行分析，首先将（a）图变换成为（b）图，然后再将电流源变换成为电压源，见图 5.17c。

两个谐振回路的耦合系数为 k，当耦合系数小于 5%时称为弱耦合，而当耦合系数大于 5%时则称为强耦合，互感 M 与耦合系数 k 的关系为：$M=k\sqrt{L_1L_2}$。

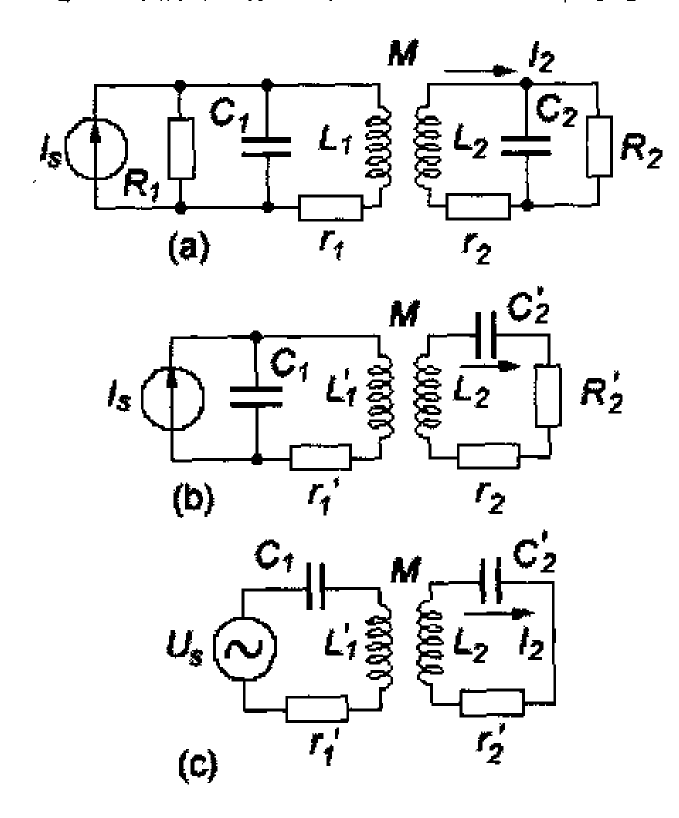


图 5.17　双调谐回路

c 图是 a 图的等效电路，它所满足的回路电压方程为：

$$\begin{cases} U_s = [r'_1 + j(\omega L_1' - \dfrac{1}{\omega C_1})] \cdot I_1 - j\omega M \cdot I_2 \\ 0 = -j\omega M \cdot I_1 + [r'_2 + j(\omega L_2 - \dfrac{1}{\omega C'_2})] \cdot I_2 \end{cases}$$

方程的解为：$I_2=\dfrac{B}{A}\cdot U_s$，其中 $B=j\omega^3 MC_1C'_2$，

$$A = (M^2-L'_1L_2)C_1C'_2\omega^4 + j(r'_1L_2 + r'_2L_1)C_1C'_2\omega^3 + (L'_1C_1 + L_2C'_2 + r'_1r'_2C_1C'_2)\omega^2 - j(r'_1C_1 + r'_2C'_2)\omega - 1$$

各元件间的变换关系为：

$$C'_2 = C_2 + \frac{1}{\omega^2 C_2 R_2^2}, \qquad r'_2 = \frac{R_2}{1+(\omega C_2 R_2)^2} + r_2$$

$$r'_1 = R_1 \cdot \frac{r_1(r_1 + R_1) + \omega^2 L^2}{(r_1 + R_1)^2 + \omega^2 L^2}, \qquad L'_1 = L_1 \cdot \frac{R_1^2}{(r_1 + R_1)^2 + \omega^2 L^2}$$

两个谐振回路本身的损耗电阻 r_1 和 r_2 可以用品质因数 Q_1 和 Q_2 来表示，电压源与电流源的关系为：$I_s = j\omega C_1 \cdot U_s$，而输出端电压与电流的关系为：

$$U_2 = I_2 \cdot \frac{1}{\frac{1}{R_2} + j\omega C_2}$$ 。

自定义 M 函数 dbtuned.m 和 dbtuned1.m 用来对上述的互感耦合谐振回路进行仿真，它们可以在 4.2 版或 5.x 版内运行，其中 dbtuned1 是根据上面的公式编写的，而 dbtuned 用来建立一个按钮界面，通过它可以方便地对各种参数进行调整。

dbtuned.m 函数的程序清单如下：

```
function [] =dbtuned (action);
global H11 H12 H21 H22 H31 H32 H41 H42 H51 H52 H61 H62 H71 H72 …
    Q1 C1 R1 Q2 C2 R2 k Kui ym Kb
if nargin<1;
  action='initialized';
end;
if strcmp (action,'initialized');
  figure ('Name','Double-tuned Circuit      BBI 2000','Num','off', …
     'Units','Normal','Position', [0.2 0.2 .7 .7]);
  v=version;
  if strcmp (v (1),'4'); whitebg ('w'); else; set (gcf,'color','w'); end;
  axes ('Units','Normal','Position', [0.1 0.1 .7 .84],'vis','off');
  uicontrol ('Style','Frame','Units','Normal','Pos', [.82 0 .2 1], …
      'Back', [.2 .4 .4]); d=.04;
  H11=uicontrol ('Style','Text','Units','Normal','Pos', …
       [.84 .95 .15 .04],'String','Q1: 100','Back', [0 1 1]);
  H12=uicontrol ('Style','Slider','Units','Normal','Position', …
       [.84 .9 .15 .04],'Back', [.8 .8 .8],'Value', .5,'Call','dbtuned Q1;');
  H21=uicontrol ('Style','Text','Units','Normal','Position', …
       [.84 .85 .15 .04],'String','C1: 210 pF','Back', [0 1 1]);
  H22=uicontrol ('Style','Slider','Units','Normal','Position', …
       [.84 .8 .15 .04],'Back', [.8 .8 .8],'Value', .5,'Call','dbtuned C1;');
  H31=uicontrol ('Style','Text','Units','Normal','Position', …
       [.84 .75 .15 .04],'String','R1: 75 k','Back', [0 1 1]);
  H32=uicontrol ('Style','Slider','Units','Normal','Position', …
       [.84 .7 .15 .04],'Back', [.8 .8 .8],'Value', .15,'Call','dbtuned R1;');
  H41=uicontrol ('Style','Text','Units','Normal','Position', …
```

```
      [.84 .65-d .15 .04],'String','Q2: 100','Back', [0 1 1]);
  H42=uicontrol ('Style','Slider','Units','Normal','Position', ...
      [.84 .6-d .15 .04],'Back', [.8 .8 .8],'Value', .5,'Call','dbtuned Q2;');

  H51=uicontrol ('Style','Text','Units','Normal','Position', ...
      [.84 .55-d .15 .04],'String','C2: 210 pF','Back', [0 1 1]);
  H52=uicontrol ('Style','Slider','Units','Normal','Position', ...
      [.84 .5-d .15 .04],'Back', [.8 .8 .8],'Value', .5,'Call','dbtuned C2;');
  H61=uicontrol ('Style','Text','Units','Normal','Position', ...
      [.84 .45-d .15 .04],'String','R2: 5 k','Back', [0 1 1]);
  H62=uicontrol ('Style','Slider','Units','Normal','Position', ...
      [.84 .4-d .15 .04],'Back', [.8 .8 .8],'Value', .25,'Call','dbtuned R2;');
  uicontrol ('Style','Popup','Units','Normal','Position', ...
      [.84 .35-d .15 .04],'String', str2mat ('Y-axis 2', ...
     'Y-axis 5','Y-axis 10','Y-axis 25','Y-axis 1'), ...
     'Back', [1 1 .6],'Call','dbtuned y-axis;');
  H71=uicontrol ('Style','Text','Units','Normal','Position', ...
      [.84 .3-2*d .15 .04],'String','k: 1 %','Back', [0 1 1]);
  H72=uicontrol ('Style','Slider','Units','Normal','Position', ...
      [.84 .25-2*d .15 .04],'Back', [.8 .8 .8], ...
     'Value', .1,'Call','dbtuned couple;');
  uicontrol ('Style','Push','Units','Normal','Position', ...
      [.84 .1 .15 .05],'String','u2/i2','Call','dbtuned u2i2;');
  uicontrol ('Style','Push','Units','Normal','Position', ...
      [.84 .02 .15 .05],'String','RESTORE','Call','dhtuned restore;');
  uimenu ('label','&Bandwidth','call','dbtuned band');
  Q1=100; Q2=100; k=.01; C1=210e-12; C2=210e-12;
  R1=75e+03; R2=5000;
  Kui=1; Kb=-1; ym=2; dbtuned1 (2);
elseif strcmp (action,'Q1');
  K=get (gco,'value'); Q1=round (K*200) +eps;
  set (H11,'string', ['Q1: ' num2str (Q1)]); dbtuned1 (ym); Kb=-1;
elseif strcmp (action,'C1');
  K=get (gco,'value'); C1=round (K*180+120);
  set (H21,'string', ['C1: ' num2str (C1) ' pF']);
  C1=C1*1e-12;   dbtuned1 (ym);   Kb=-1;
elseif strcmp (action,'R1');
  K=get (gco,'value'); R1=round (K*500);
  set (H31,'string', ['R1: ' num2str (R1) ' k']);
```

```
    R1 = R1 * 1e + 03;    dbtuned1 (ym);    Kb = -1;
  elseif strcmp (action,'Q2');
    K = get (gco,'value'); Q2 = round (K * 200)  + eps;
    set (H41,'string', ['Q2: ' num2str (Q2)]); dbtuned1 (ym); Kb = -1;
  elseif strcmp (action,'C2');
    K = get (gco,'value'); C2 = round (K * 180 + 120);
    set (H51,'string', ['C2: ' num2str (C2) ' pF']);
    C2 = C2 * 1e - 12; dbtuned1 (ym);    Kb = -1;
  elseif strcmp (action,'R2');
    K = get (gco,'value'); R2 = round (K * 2000)  * .01;
    set (H61,'string', ['R2: ' num2str (R2) ' k']);
    R2 = R2 * 1e + 03;    dbtuned1 (ym);    Kb = -1;
  elseif strcmp (action,'couple');
    K = get (gco,'value'); k = K^2; k = round (k * 1000)  * .001;
    set (H71,'string', ['k: ' num2str (k * 100) ' %']); dbtuned1 (ym); Kb = -1;
  elseif strcmp (action,'restore');
    Q1 - 100;    set (H11,'string','Q1: 100');    set (H12,'valuc', .5);
    C1 = 210e - 12; set (H21,'string','C1: 210 pF'); set (H22,'value', .5);
    R1 = 75e + 03;    set (H31,'string','R1: 75 k');    set (H32,'value', .15);
    Q2 = 100;    set (H41,'string','Q2: 100');    set (H42,'value', .5);
    C2 = 210e - 12; set (H51,'string','C2: 210 pF'); set (H52,'value', .5);
    R2 = 5e + 03;    set (H61,'string','R2: 5 k');    set (H62,'value', .25);
    k = 0.01;    set (H71,'string','k: 1 %');    set (H72,'value', .1);
    ym = 2;    Kui = 1;    dbtuned1 (2);    Kb = -1;
  elseif strcmp (action,'y - axis');
    y = [2 5 10 25 1]; K = get (gco,'value'); ym = y (K) dbtuned1 (ym); Kb = -1;
  elseif strcmp (action,'u2i2');
    Kui = -Kui; dbtuned1 (ym);    Kb = -1;
  elseif strcmp (action,'band');
    Kb = -Kb; dbtuned1 (ym, Kb);
  end;
```

dbtuned1.m 函数的程序清单如下：

```
function [f, y] = dbtuned1 (ym, Kb)
if nargin<2; Kb = -1; end;
global C1 C2 R1 R2 Q1 Q2 k Kui
f = (100: 900) * 1e + 03; w = 2 * pi * f;    f0 = 465e + 03;
L1 = 5.5785e - 04;    L2 = 5.5785e - 04;    M = k * sqrt (L1 * L2);
```

```
r1=1/ (2*pi*f0*C1*Q1);     r2=1/ (2*pi*f0*C2*Q2);
C22=C2+1./ (w.^2*C2*R2^2); r22=R2./ (1+ (w*C2*R2) .^2) +r2;
r11=R1* (r1* (r1+R1) +w.^2*L1^2) ./ ( (r1+R1) ^2+w.*w*L1*L1);
L11=L1*R1^2./ ( (r1+R1) ^2+w.*w*L1*L1);
B=j*w.^3*M*C1.*C22;
A= (M^2-L11*L2) *C1.*C22.*w.^4;
A=A+j* (r11*L2+r22.*L11) *C1.*C22.*w.^3;
A=A+ (L11*C1+L2*C22+r11.*r22*C1.*C22) .*w.^2;
A=A-j* (r11*C1+r22.*C22) .*w-1;
is=0.001;       us=is./ (j*w*C1);
i2=B./A.*us;    u2=i2./ (1/R2+j*w*C2); f=f*.001;
if Kui==1;
  u2=abs (u2); [ymax I] =max (u2); f0=f (I); I=find (u2>ymax*.7071);
  y1=u2 (I (1)); y2=u2 (max (I)); f1=f (I (1)); f2=f (max (I));
  plot (f, u2,'b', [100 900], [1 1] *.7071*ymax,'b:'); ylabel ('u2 (V)');
elseif Kui==-1;
  i2=abs (i2) *1000; [ymax I] =max (i2); f0=f (I);
  I=find (i2>ymax*.7071);
  y1=i2 (I (1)); y2=i2 (max (I)); f1=f (I (1)); f2=f (max (1));
  plot (f, i2,'b', [100 900], [1 1] *.7071*ymax,'b:'); ylabel ('i2 (mA)');
end;
grid; zoom xon; axis ( [100 900 0 ym]); xlabel ('Frequency (kHz)');
title ( [num2str (ymax) ' (' num2str (f0) ' kHz)']);
if Kb>0;
  text (820, ymax*.75,'-3dB'); f0= (f1+f2) /2;
  text (f0, ymax*1.1, num2str (round (f0)),'hor','center');
  text (f1-20, ymax*.75, num2str (round (f1)),'hor','right');
  text (f2+20, ymax*.75, num2str (round (f2)));
end;
```

使用自定义 M 函数 dbtuned.m 和 dbtuned1.m 用来对互感耦合谐振回路进行仿真，可以清楚地了解每个元件参数对整个谐振回路的影响，使用者可以观察输出电压或回路电流随频率的变化。改变耦合系数对谐振回路的影响是十分明显的，在耦合比较弱的时候，幅频特性呈单峰；而在耦合比较强的情况下，幅频特性呈双峰。初级回路电容 C_1 对谐振频率的影响比较大，而次级回路电容 C_2 对谐振曲线的形状影响比较大。使用 bandwidth 菜单还可以显示出与谐振回路通频带有关的各种参数。图 5.18 为一个典型的仿真结果，其中图形上方的数字表示峰值的大小和频率。

从图 5.18 中给出的仿真结果可以清楚地看出双调谐回路的特点，由于幅频特性曲线出现了双峰，因此其通频带是相当宽的。仿真的结果表明，该谐振回路处于双峰状态时，其通

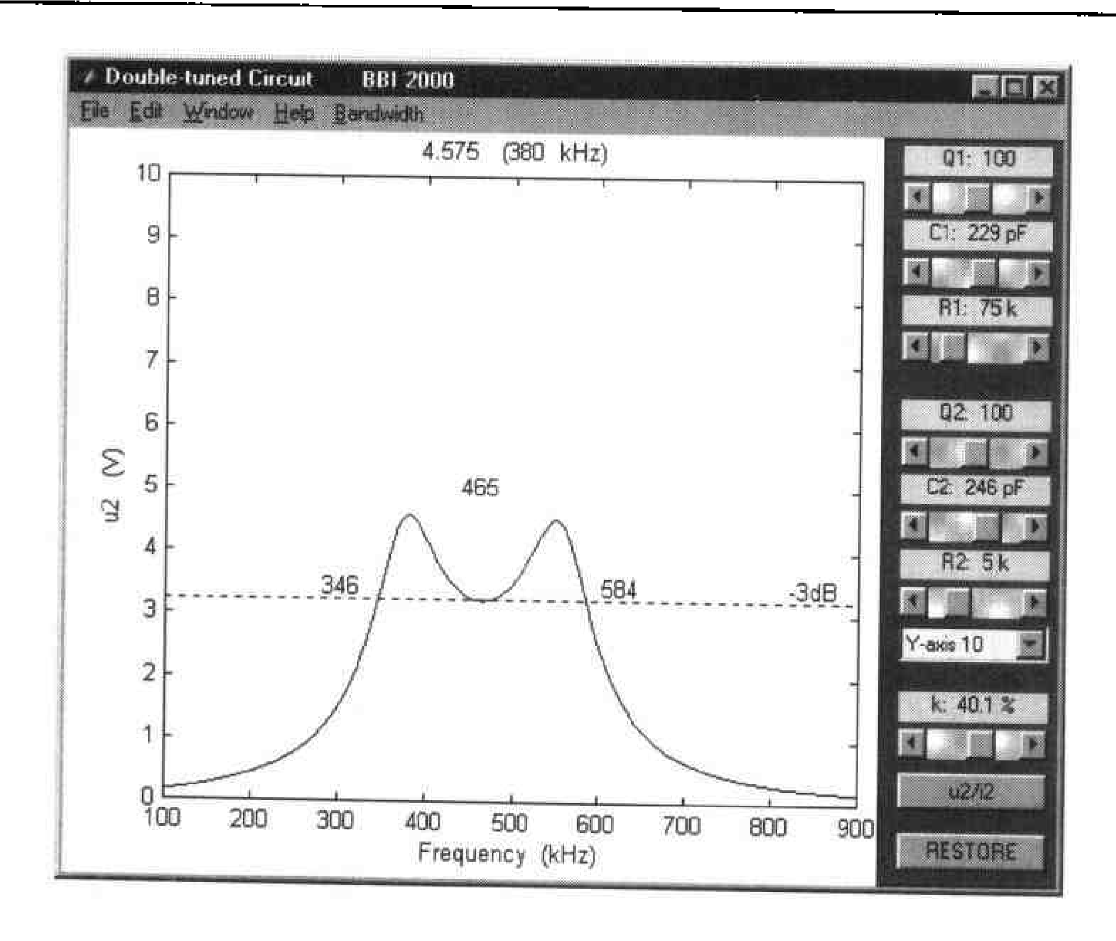


图5.18 双调谐回路的仿真

频带宽度为238MHz（按3dB带宽计算），相对带宽高达51.18%，而处于单峰状态时，相对带宽仅为3.01%。

5.5 小信号放大器

小信号放大器是电子线路的重要组成部分之一，由于它工作在晶体管的线性区域之内，因此又称为线性放大器。使用MATLAB可以仿真小信号放大器的各种参数，如电压增益、输入阻抗、输出阻抗、频率响应等等。

5.5.1 晶体三极管的等效电路

常见的晶体三极管等效电路有：低频h参数、共基极T型高频等效电路、混合π型高频等效电路，它们通常用于分析各种小信号晶体管放大器的特性。

共发射极h参数的等效电路如图5.19（a）所示，它适用于对低频放大器进行分析。另外，还存在着一种简化的h参数等效电路，其中忽略晶体管内部的电压反馈系数h_{re}。共发射极的h参数与各电压电流的关系为：$\begin{bmatrix} v_b \\ i_c \end{bmatrix} = \begin{bmatrix} h_{ie} & h_{re} \\ h_{fe} & h_{oe} \end{bmatrix} \begin{bmatrix} i_b \\ v_c \end{bmatrix}$。

共基极T型高频等效电路如图5.19（b）所示，它适用于对共基极高频放大电路进行分析，工作频率可高达100MHz以上。

混合π型高频等效电路如图5.19（c）所示，它适用于分析共发射极高频放大电路，在

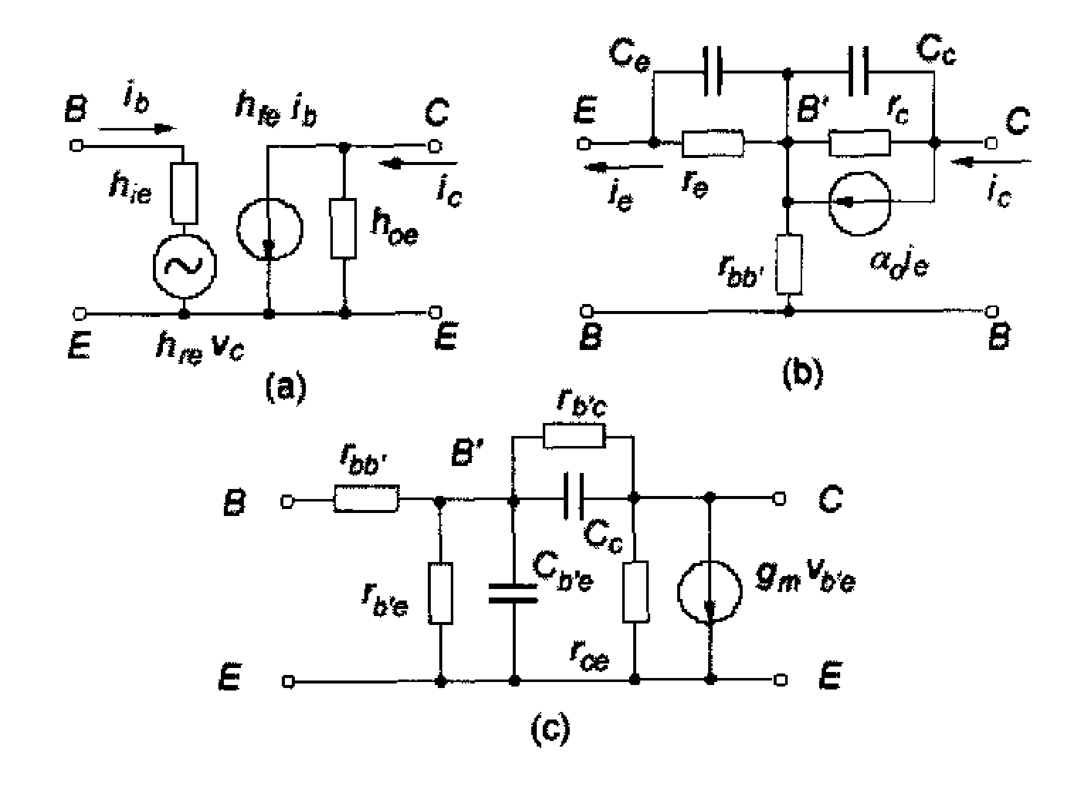


图 5.19　晶体三极管的等效电路

较宽的频率范围之内，等效电路的参数与工作频率无关。另外还存在着简化的混合 π 型高频等效电路，其中 $r_{b'e}$和 r_{ce}处于开路状态。

5.5.2　共发射极放大电路

共发射极放大电路是一种使用得最为广泛的放大电路形式，其特点是电压增益和电流增益都比较高。自定义 M 函数 amplif1.m 就用来仿真共发射极放大电路，使用它可以计算该放大器的直流参数和交流参数（频率在 1000Hz 左右的中间频率），该放大器的电路见图 5.20。

MATLAB 的特点之一就是适合进行线性代数运算，因此无论在分析直流参数或分析交流参数时，都可以采用基尔霍夫（Kirchhoff）定律，即流入一个节点电流的代数和为零。这样很容易建立起电路方程，然后采用矩阵求逆的方式求出电压和电流的具体数值，进一步便可得到该放大器的各项参数。在分析共发射极放大的交流参数时，采用的晶体管模型是低频 h 参数等效电路。一般来说，每个晶体管可以用三个节点来表示，它们分别是：基极、集电极和发射极。在计算交流参数过程之中，忽略各电容器的容抗。

amplif1.m 函数的用法是：

》[Av, Zi, Zo, Ie, Vb, Vc, vo] =amplif1 (Rb1, Rb2, Rc, Re, RL, h, Rs, vs, beta, Ec, Kp)

输入参数：　　　（其中电阻的单位均为 Ω）

h= [hie hre; hfe hoe] 晶体管的 h 参数，　　beta 晶体管的直流放大系数

Ec　电源电压（V），　　参数 Kp=1 表示硅管，Kp=2 表示锗管

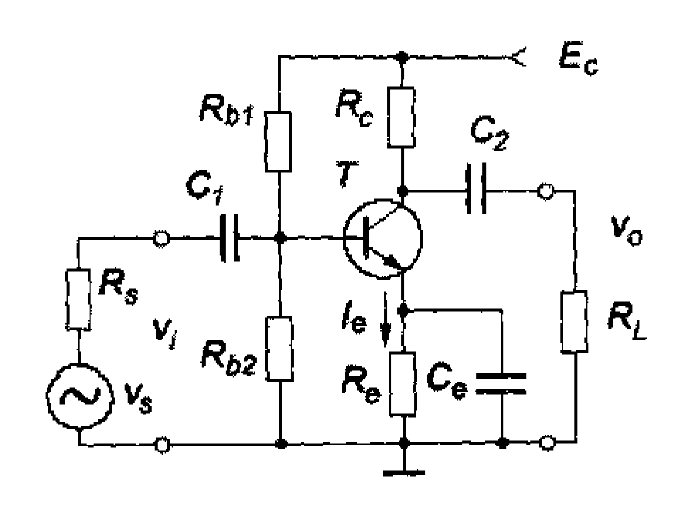


图 5.20 共发射极放大电路

v 信号源的开路电压

输出参数：

Av 电压放大倍数， Zi 输入阻抗（Ω）， Zo 输出阻抗（Ω）

Ie 集电极电流（mA）， Vb 基极电压（V）， Vc 集电极电压（V）

vs 放大器输出电压（V）

amplif1.m 的程序清单如下：

```
function [Av, Zi, Zo, Ie, Vb, Vc, vo] =amplif1 (Rb1, Rb2, Rc, Re, RL, …
                    h, Rs, vs, beta, Ec, Kp);
%
%  Usage: [Av, Zi, Zo, Ie, Vb, Vc, vo] =amplif1 (Rb1, Rb2, Rc, Re, RL, …
%                    [hie hre; hfe hoe], Rs, vs, heta, Ec, Kp);
%
% [Av, Zi, Zo, Ie, Vb, Vc, vo] =amplif1 (24e+03, 6e+03, 2e+03, 1.5e+03,
%         6e+03, [1.2e+03 3.37e-4; 50 27.1e-6], 1e+03, .01, 50, 12, 1)
%    BBI 2000
if nargin<11;    Kp=1;       end;                    %缺省参数
if nargin<10;    Ec=12;       end;
if nargin<9;     beta=50;    end;
if nargin<8;     vs=10e-03;     end;
if nargin<7;     Rs=1e+03;     end;
if nargin<6;     h= [1.2e+03 3.37e-4; 50 27.1e-6]; end;
if nargin<5;     RL=6e+03;     end;
if nargin<4;     Re=1.5e+03;     end;
if nargin<3;     Rc=2e+03;     end;
if nargin<2;     Rb2=6e+03;     end;
```

```
if nargin<1;    Rb1=24e+03;    end;
Rb=Rb1*Rb2/(Rb1+Rb2); YL=(Rc+RL)/(Rc*RL);
Rs1=Rs*Rb/(Rs+Rb);
Zi=b(1,1)-h(2,1)*h(1,2)/(YL+b(2,2));    %交流参数
Z1=Zi*Rb/(Zi+Rb);    vb=vs*Z1/(Z1+Rs);    ib=vb/Zi;
Ro=1/h(2,2);    ic=h(2,1)*ib*Ro/(Ro+1/YL);
vo=-ic/YL;    Av=vo/vb;    format short;
Yo=b(2,2)-h(2,1)*h(1,2)/(h(1,1)+Rs1)+1/Rc;    Zo=1/Yo;
Zi=round(Zi); Zo=round(Zo);    Av=round(Av*10)*.1;
if Kp==1; Vbe=.6; ns='Si'; else; Vbe=.2; ns='Ge'; end;
A=[(Rb1+Rb2)/Rb2 Rb1; 1 -(1+beta)*Re];    %直流参数
B=inv(A)*[Ec Vbe]';
Vb=B(1); Ib=B(2); Ie=(1+beta)*Ib; Vc=Ec-beta*Ib*Rc;
Vb=round(Vb*10)*.1; Vc=round(Vc*10)*.1;
Ie=round(Ie*1e+04)*.1;
```

使用缺省参数时的运算结果为：

```
》[Av, Zi, Zo, Ie, Vb, Vc, vo] =amplif1
Av = -61.3000                     Vb=2.3000
Zi =1176                          Vc=9.8000
Zo =1928                          Vo= -0.2977
Ie =1.1000
```

5.5.3 直接耦合放大器

在两个或三个晶体管之间进行直接耦合的放大器称为直接耦合放大器，它多用作音响系统中的前置放大器、录音机内的磁头放大器。直接耦合放大器的主要特点是直流工作点稳定，电压增益高，图 5.21 是一个典型的直接耦合放大电路，它由三个晶体管组成，第一级为低噪声放大，第二级为高增益放大，第三级为射随器，整个放大器的电压增益由负反馈电路确定。由于采用了串联电压负反馈，同时又使用了射随器，因此该电路的具有较高的输入阻抗和较低的输出阻抗。

自定义 M 函数 amplif2.m 用来分析图 5.21 所示的直接耦合放大器的交流参数和直流参数，其用法是：

[A, Zi, Zo, Vb, Ie, E] =amplif2 (Rb1, Re1, Rc1, R1, R2, Rc2, Re3, Rf, h, vs, beta, Ec, Ed, Kp);

输入参数：　　(其中电阻的单位均为 Ω)

h=[hie hre; hfe hoe] 晶体管的 h 参数，　beta 晶体管的直流放大系数

Ec　电源电压 (V)，　Ed 第一级的电源电压 (V)

参数 Kp=1 表示硅管，Kp=2 表示锗管

输出参数：

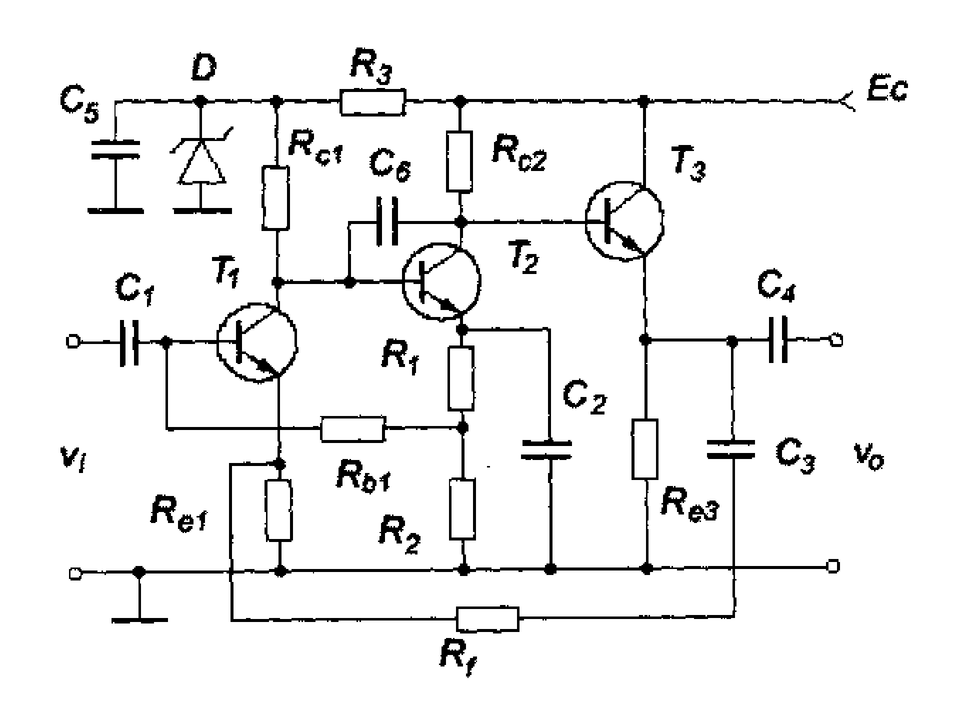


图 5.21 直接耦合放大器

A= [Av Av0 Av1 Av2] Av 电压放大倍数， Av0 开环电压放大倍数
Av1 第一级电压放大倍数，Av2 第二级电压放大倍数
Zi 输入阻抗（Ω）， Zo 输出阻抗（Ω）
Vb= [Vb (1) Vb (2) Vb (3)] 三个晶体管的基极电压（V）
Ie= [Ie (1) Ie (2) Ie (3)] 三个晶体管的发射极电流（mA）

amplif2.m 的程序清单如下：

```
function [Av, Zi, Zo, Vb, Ie, E, vo] = amplif2 (Rb1, Re1, Rc1, R1, R2, …
         Rc2, Re3, Rf, h, vs, beta, Ec, Ed, Kp);
%
%
%
%   BBI 2000
if nargin<14; Kp=1;        end;
if nargin<13; Ed=15;        end;
if nargin<12; Ec=24;        end;
if nargin<11; beta=50;       end;
if nargin<10; vs=1e-03;       end;
if nargin<9; h= [1.2e+03 3.37e-4; 80 27.1e-6]; end; % [hie hre; hfe hoe]
if nargin<8; Rf=33e+03;       end;
if nargin<7; Re3=3.3e+03;       end;
if nargin<6; Rc2=18e+03;       end;
```

```
if nargin<5; R2=3.9e+03;      end;
if nargin<4; R1=130;      end;
if nargin<3; Rc1=100e+03;      end;
if nargin<2; Re1=100;    end;
if nargin<1; Rb1=1000e+03;      end;
hie=h(1,1); hfe=h(2,1); hoe=h(2,2); Rc=hie*Rc1/(hie+Rc1);
T=[hoe+1/Re1 -hoe -1-hfe; -hoe hoe+1/Rc hfe; 1 0 hie];
V=inv(T)*[0 0 vs]';   v2=V(2);   ib2=v2/hie;   Av1=v2/vs;
Zi=vs/V(3);
Re=Re3/hoe/(Re3+1/hoe); Rc=Rc2/hoe/(Rc2+1/hoe);
T=[1/Re -1-hfe; 1 hie+Rc];
V=inv(T)*[0 -hfe*Rc*ib2]'; Av2=V(1)/v2;
Av0=V(1)/vs; % [Av0 Av1 Av2 Av1*Av2]
Zo=V(1)/((1+hfe)*hfe*Rc*ib2/(Rc+hie)); Zo=abs(Zo);
B=Re1/(Rf+Re1); F=1+Av0*B;
Av=Av0/F;   Zi=Zi*F; Zi=Zi*Rb1/(Zi+Rb1);   Zo=Zo/F;   vo=Av*vs;
Av=[Av Av0 Av1 Av2];
if Kp==1; Vbe=.7; ns='Si'; else; Vbe=.2; ns='Ge'; end;
Z=[Rb1+R2+(1+beta)*Re1 -(1+beta)*R2;
beta*Rc1-R2 Rc1+(1+beta)*(R1+R2)];
Ib=inv(Z)*[-Vbe Ed-Vbe]';
I1=(1+beta)*Ib(1); I2=(1+beta)*Ib(2);
I3=(1+beta)*(Ec-Vbe-Rc2*beta*Ib(2))/(Rc2+(1+beta)*Re3);
I=[I1 I2 I3]; I=round(I*1e+04)*.1;
V=[Re1*I1 (R1+R2)*I2 Re3*I3]+Vbe; V=round(V*10)*.1;
Zi=round(Zi);        Zo=round(Zo);        Av=round(Av);
E=[Ec Ed];           Vb=V;       Ie=I;       format short;
```

使用缺省参数时的运算结果为：

```
》[Av, Zi, Zo, Vb, Ie, E] =amplif2
Av =
      317        7766        -10        -765
Zi =
      180938
Zo =
      6
Vb =
      0.7000        3.5000        10.6000
Ie =
```

```
        0.1000        0.7000        3.0000
E =
        24        15
```

5.5.4 差分放大器

差分放大器又称为差动放大器，由于它比较好地解决了零点漂移的问题，因此多作为直流放大器来使用。差分放大器有两个输入端口和两个输出端口，于是可以分为双端输入双端输出、双端输入单端输出、单端输入双端输出、单端输入单端输出等几种形式。图 5.22 所示的是一种单端输入双端输出的差分放大器，它是模拟卫星接收机内的视频放大器，为了降低输出阻抗使用了射随器，而为了稳定工作点，使用了晶体管 T5 作为恒流源，另外恒流源的交流阻抗是相当大的，这样有利于改善差分放大器的特性。

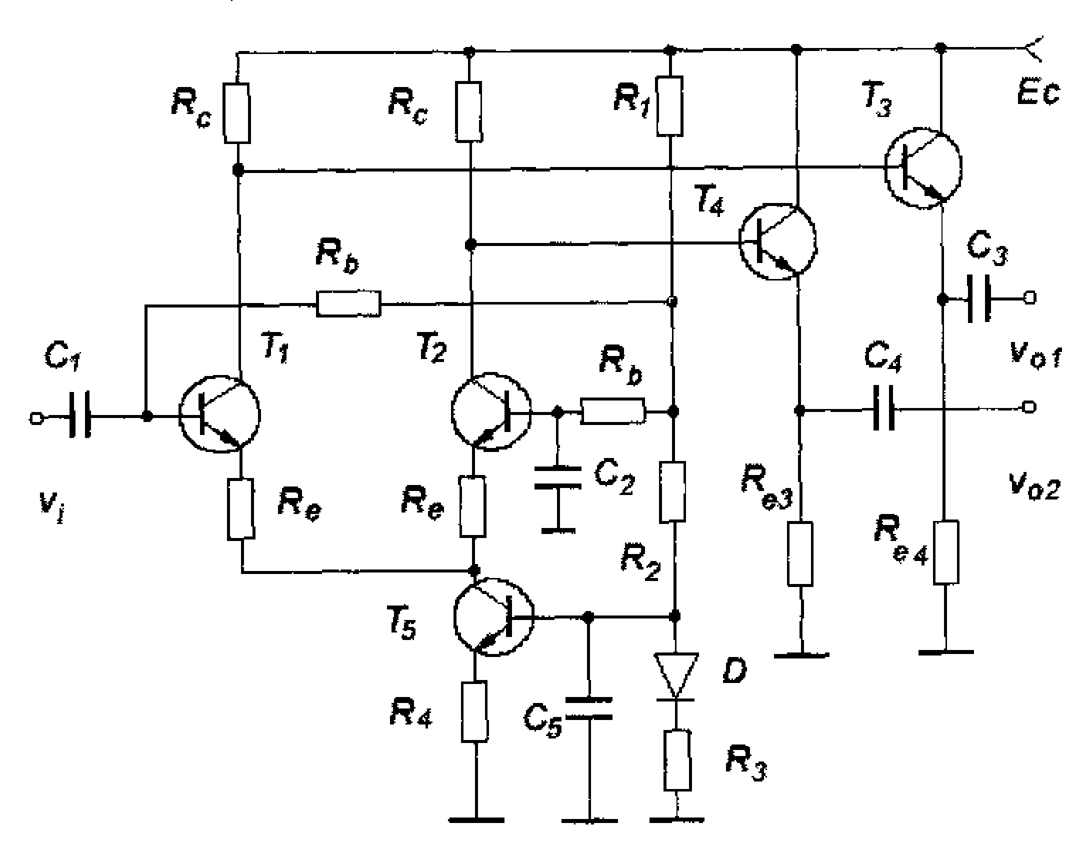


图 5.22 单端输入双端输出的差分放大器

由于射随器的输入阻抗很大，因此在分析差分放大器交流参数的过程中，完全可以忽略射随器输入阻抗对前一级的影响，这样该放大器的交流等效电路就可以用 7 个节点来描述，这 7 个节点是：晶体管 T1 的基极、发射极和集电极，晶体管 T2 的基极、发射极和集电极，晶体管 T5 的集电极。T1 的基极为放大器的输入端，而 T2 的基极在交流上是接地的，因此节点方程有 5 个，另外在加上两个晶体管的基极电流方程，这样一共就得到 7 个方程。使用矩阵的方法求解这些方程，就可以得到差分放大器的各项交流参数。

自定义 M 函数 amplif3.m 用来分析图 5.22 所示的差分放大器的各项交流参数和直流参

数，其用法是：

》[Av, Zi, Zo, V, I] = amplif3 (Rb, Rc, Re, R1, R2, R3, R4, Re3, Zee, h, vs, beta, Ec, Kp)

输入参数：　　(其中电阻的单位均为 Ω)

Zee　恒流源的等效交流阻抗，

h = [hie hre; hfe hoe] 晶体管的 h 参数，　beta 晶体管的直流放大系数

Ec　电源电压 (V)，　参数 Kp = 1 表示硅管，Kp = 2　表示锗管

输出参数：

Av = [Av1 Av2]，　Av1　1 端的电压放大倍数，　Av2　2 端的电压放大倍数

Zi　输入阻抗 (Ω)，　Zo　输出阻抗 (Ω)

V = [Vb5 Vb2 Vb4]，　晶体管 T5、T2、T2 的基极电压 (V)

Ie = [Ie1 Ie4]，　晶体管 T1、T4 的发射极电流 (mA)

amplif3.m 的程序清单如下：

```
function [Av, Zi, Zo, V, I, vo] = amplif3 (Rb, Rc, Re, R1, R2, R3, R4, …
                                    Re3, Zee, h, vs, beta, Ec, Kp);
%
%  Usage: [Av, Zi, Zo, V, I, vo] = amplif3 (Rb,Rc,Re,R1,R2,R3,R4,…
                                    Re3, Zee, b, vs, beta, Ec, Kp);
%
%  amplif3 (4700, 6.8e+03, 68, 40e+03, 20e+03, 680, 47, 2200);
%  BBI 2000
if nargin<14; Kp=1;          end;
if nargin<13; Ec=12;         end;
if nargin<12; beta=50;       end;
if nargin<11; vs=10e-03;     end;
if nargin<10;
      h= [1.2e+03 3.37e-4; 100 27.1e-6]; % [hie hre; hfe hoe]
end;
if nargin<9; Zee=1e+05;      end;
if nargin<8; Re3=3300;       end;
if nargin<7; R4=47;          end;
if nargin<6; R3=680;         end;
if nargin<5; R2=2400;        end;
if nargin<4; R1=4800;        end;
if nargin<3; Re=68;          end;
if nargin<2; Rc=560;         end;
if nargin<1; Rh=4700;        end;
hie=h (1, 1); hfe=h (2, 1); hoe=h (2, 2);          %计算交流参数
```

```
A= [1 0 0 0 0 hie 0; 0 0 1 0 0 0 hie;];
A= [A; (1+Re*hoe) -Re*hoe 0 0 -1 - (1+hfe) *Re 0];
A= [A; 0 0 (1+Re*hoe) -Re*hoe -1 0 - (1+hfe) *Re];
A= [A; Zee 0 Zee 0 - (2*Zee+Re) 0 0];
A= [A; Rc*hoe - (1+Rc*hoe) 0 0 0 -hfe*Rc 0];
A= [A; 0 0 Rc*hoe - (1+Rc*hoe) 0 0 -hfe*Rc];
V=inv (A) * [vs 0 0 0 0 0 0]';
vo= [V (2) V (4)];              Av=vo/vs;              Av=round (Av*10) *.1;
Rb1=1/ (1/Rb+1/R2+1/R1);        Zi=vs/V (6);           Zi=Zi*Rb1/ (Zi+Rb1);
Zo= (Rc+hie) / (1+hfe);         Zo=round (Zo);         Zi=round (Zi);
if Kp==1; Vbe=.8; ns='Si'; else; Vbe=.2; ns='Ge'; end;
                                                  % 计算直流参数
A= [ (R1+R2+R3) /R3 R1+R2 R1; 1 - (1+beta) *R4 0; 0 beta -2* (1+be-
ta)];
B=inv (A) * [Ec+ (R1+R2) /R3*Vbe Vbe 0]';        format short;
V= [B (1) Ec-R1* ((B (1) -Vbe) /R3+B (2) +B (3)) -Rb*B (3) Ec-be-
ta*B (3) *Rc];
I= [B (3) * (1+beta) (V (3) -Vbe) /Re3]; I=round (I*1e+04) *.1;
```

使用典型参数时的运算结果为：

```
[Av, Zi, Zo, V, I] =amplif3 (4700, 6.8e+03, 68, 40e+03, 20e+03, 680, 47,
2200, 1E+05)
Av =
    -41.0000    40.9000
Zi =
    2783
Zo =
    79
V =
    0.8915    4.2570    5.6401
I =
    1.0000    2.2000
```

5.5.5 阻容耦合音频放大器的频率响应

阻容耦合音频放大器的电路如图 5.20 所示，自定义 M 函数 amplif1.m 在分析该放大器交流参数时未考虑电容的容抗，而分析该电路的频率响应时不能忽略各个电容的影响。由于音频的频率范围在 20－20000Hz 之间，因此在分析音频放大器时可以采用低频 h 参数，同时忽略晶体管内部的反馈，于是阻容耦合音频放大器的交流等效电路如图 5.23 所示，显然这是一个典型的两端口网络，分析其特性使用 A 参数较为方便。

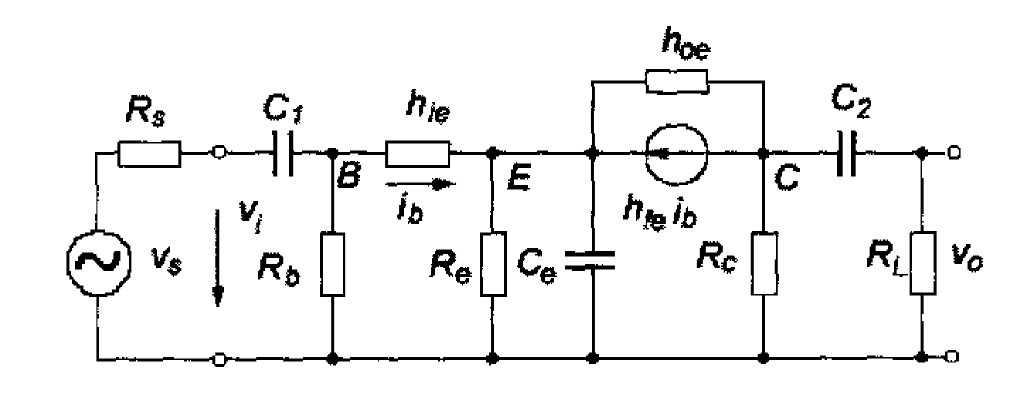


图5.23 阻容耦合音频放大器的交流等效电路

将晶体管的 h 参数转换为 Z 参数后，有$\begin{bmatrix} v_b \\ v_c \end{bmatrix} = \begin{bmatrix} h_{ie} & 0 \\ -h_{fe}/h_{oe} & 1/h_{oe} \end{bmatrix} \begin{bmatrix} i_b \\ i_c \end{bmatrix}$，考虑到发射极的电阻和电容，这相当于两个串联的两端口网络，即总的 Z 参数为晶体管的 Z 参数与发射极电阻电容的 Z 参数之和，然后再将 Z 参数转换成为 A 参数。这样就可以使用 T 型网络 A 参数相乘的方式求出整个等效电路的 A 参数，进一步即可得到其幅频特性、相频特性和输入阻抗。

自定义 M 函数 amplif4.m 根据图 5.23 所示的等效电路来分析阻容耦合音频放大器的各项交流参数和直流参数，由于使用了符号运算，因此它是工作在 5.x 版环境下的，其用法是：

》[H, Zi] =amplif4 (Ce, C1, C2, Rb1, Rh2, Rc, Re, RL, h, Rs);

输入参数：　　(其中电阻的单位均为 Ω，电容的单位均为 F)

C1	基极耦合电容，	C2	集电极耦合电容
Ce	发射极旁路电容，	Rs	信号源内阻
h= [hie hre; hfe hoe]		晶体管的 h 参数	

输出参数：

H=vo/vi　放大器的转移函数，　Zi　输入阻抗

amplif4.m 的程序清单如下：

```
function [H, Zi] =amplif4 (Ce, C1, C2, Rb1, Rb2, Rc, Re, RL, h, Rs);
%   Frequency Responce of Common Emitter Audio Amplifer,    BBI 2000
%
%   Usage: [H, Zi] =amplif4 (Ce, C1, C2, Rb1, Rb2, Rc, Re, RL, h, Rs);
%
%    [H, Zi] =amplif4 (Ce, C1, C2, Rb1, Rb2, Rc, Re, RL, h, Rs)
if nargin<10; Rs=1e+03;          end;
if nargin<9; h= [1.2e+03 3.37e-4; 50 27.1e-6]; end; % [hie hre; hfe hoe]
if nargin<8; RL=6e+03;           end;
if nargin<7; Re=1.5e+03;         end;
if nargin<6; Rc=2e+03;           end;
```

```
if nargin<5; Rb2=6e+03;          end;
if nargin<4; Rb1=24e+03;         end;
if nargin<3; C2=20e-06;          end;
if nargin<2; C1=20e-06;          end;
if nargin<1; Ce=200e-06;         end;
syms s;
hie=h (1, 1);   hfe=h (2, 1);   hoe=h (2, 2);
zt= [hie 0; -hfe/hoe 1/hoe];   ze=Re/ (1+s*Re*Ce); ze=ones (2, 2) *ze;
Z=zt+ze;     A= [Z (1, 1) det (Z); 1 Z (2, 2)] /Z (2, 1);     % BC之间的
A矩阵
Rb=Rh1*Rb2/ (Rb1+Rb2);
A= [1 Rs+1/s/C1; 0 1] * [1 0; 1/Rb 1] *A* [1 0; 1/Rc 1] *[1 1/s/C2;0 1];
A=A* [1 0; 1/RL 1]; Zi=A (1, 1) /A (2, 1) -Rs;
f=logspace (1, 5, 101);
[b, a] =numden (Zi); b=sym2poly (b);   a=sym2poly (a);
Zi=freqs (b, a, 2*pi*f); k=max (abs ( (Zi+Rs) ./Zi));
H=k/A (1, 1);      [b, a] =numden (H);
b=sym2poly (h);   a=sym2poly (a);   H=freqs (b, a, 2*pi*f);
Av=20*log10 (ahs (H)); Avm=round (max (Av) *10) *.1;
subplot (211); semilogx (f, Av);   grid;   zoom xon;%幅频特性
xlabel ('frequency (Hz)');   ylabel ('Av (dB)');
title ( ['Av_m_a_x = ' num2str (Avm) ' (dB)']);
subplot (212); semilogx (f, real (Zi), f, imag (Zi)); grid;   zoom xon;
xlabel ('frequency (Hz)');   ylabel ('Zi (Ohm)');
set (gcf,'units','pix','pos', [200, 120, 560, 420], …
  'name','Common Emitter Amplifer, BBI 2000','num','off');
```

使用缺省参数时的运算结果见图 5.24。从仿真的结果可以看出，影响音频放大器低频端增益的主要因素是发射极的旁路电容 Ce，而耦合电容对低频端增益的影响是比较小的。在设计高保真放大器时，为了得到比较平坦的幅频特性曲线，Ce 的数值通常应该在 500μF 以上；另外还可以采用电流负反馈方式，索性去除 Ce，这样发射极电阻 Re 的数值应该取得比较小，以免增益下降得过多。

5.5.6 共发射极放大电路的高频频率响应

分析共发射极放大电路的高频频率响应，晶体管应该采用混合 π 型高频等效电路，这样整个共发射极放大器的高频等效电路如图 5.25 所示。表征一个晶体管高频工作特性的参数主要有：特征频率 f_T，集电极电容 C_C，集电极工作电流 Ic，其它参数可以由上述参数和低频 h 参数得到。

跨导
$$g_m = I_c\ (mA)\ /26 \qquad (S)$$

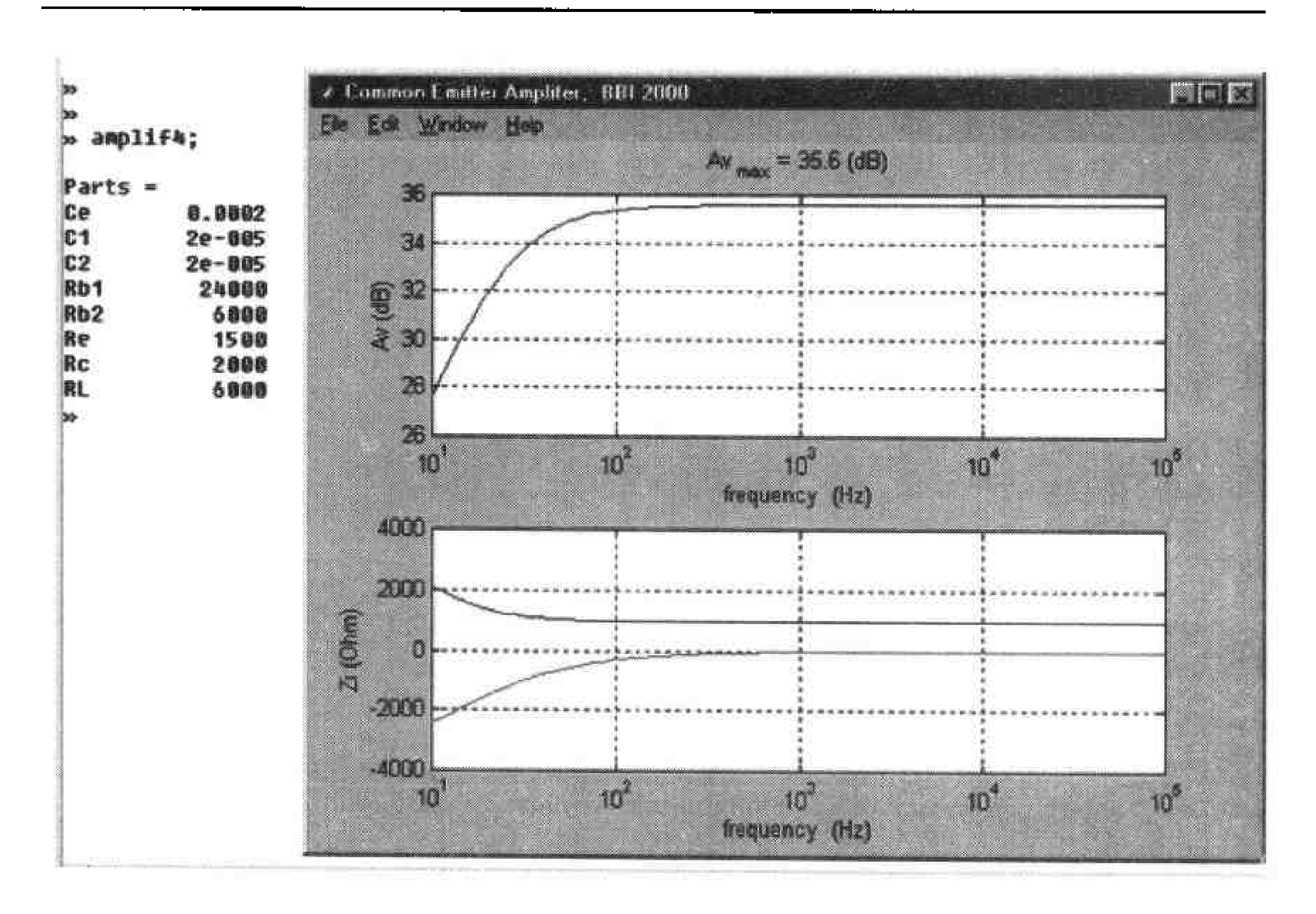


图 5.24 阻容耦合音频放大器的幅频特性和输入阻抗曲线

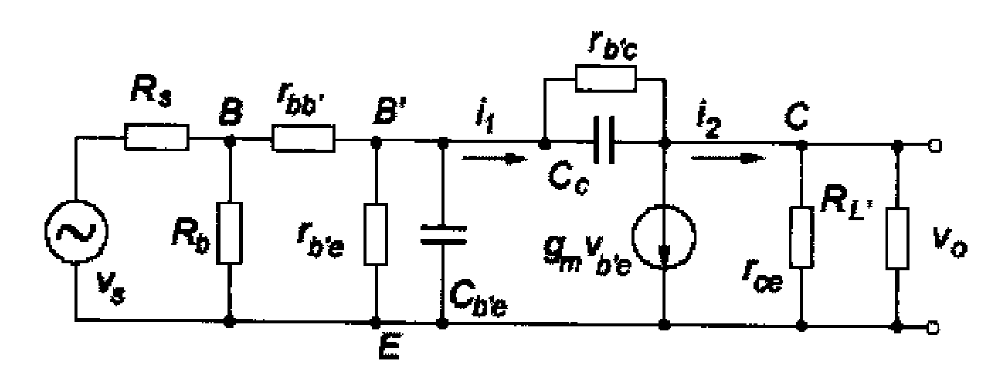


图 5.25 共发射极放大器的高频等效电路

发射结电容 $C_{b'e}=\dfrac{g_m}{2\pi\cdot f_T}$ (F)

发射极交流电阻 $r_{b'e}=h_{fe}/g_m$ (Ω)

基区体积电阻 $r_{bb'}=h_{ie}-r_{b'e}$ (Ω)

集电极交流电阻 $r_{b'c}$的数值通常在 1M 以上。

由于共发射极放大电路为两端口网络，故分析其特性使用 A 参数是非常方便的。晶体管等效电路中 B′和 C 之间的 A 矩阵可以由以下的方程组求出：

$$\begin{cases} i_1-i_2=g_m v_{b'e} \\ v_{b'e}-v_c=Zc\cdot i_1 \end{cases},\qquad \text{于是有}\quad A=\frac{1}{1-g_m Z_c}\begin{bmatrix}1 & Z_c\\ g_m & 1\end{bmatrix}$$

自定义M函数amplif5.m根据图5.25所示的等效电路来分析共发射极放大器的高频频率响应，由于使用了符号运算，因此它是工作在5.x版环境下的，其用法是：

》H=amplif5（Cc，ft，Ic，Rb1，Rb2，Rc，RL，h，Rs，rbc)；

输入参数：（其中电阻的单位均为Ω，电容的单位均为F）

Cc　集电极电容，　ft　晶体管的特征频率

Ic　集电极电流，单位为mA

Rb1和Rb2　基极偏流电阻（见图5.20)，两者并联之后为Rb

Rc　集电极电阻，　RL　负载电阻

Rs　信号源内阻，　rbc　集电极交流电阻

h=［hie hre；hfe hoe］　晶体管的h参数

输出参数：

H=vo/vs　放大器的转移函数

amplif5.m的程序清单如下：

```
function [H] =amplif5 (Cc, ft, Ic, Rb1, Rb2, Rc, RL, Rs, h, rbc);
 % Frequency Responce of Common Emitter Amplifer,  BBI 2000
 %
 % Usage: [H, Zi] =amplif5 (Cc, ft, Ic, Rb1, Rb2, Rc, RL, Rs, h, rbc);
 %
 % H=amplif5 (3e-012, 100e+06, 2.5, 24e+03, 6e+03, 2000, 6000, 1000, h,
rbc);
 if nargin<10; rbc=5e+06;          end;
 if nargin<9; h= [1200 3.37e-4; 50 27.1e-6]; end; % [hie hre; hfe hoe]
 if nargin<8; Rs=1e+03;            end; %Rs=1e+03; 75;
 if nargin<7; RL=6e+03;            end;
 if nargin<6; Rc=2e+03;            end;
 if nargin<5; Rb2=6e+03;           end;
 if nargin<4; Rb1=24e+03;          end;
 if nargin<3; Ic=2.5;              end;
 if nargin<2; ft=100e+06;          end;
 if nargin<1; Cc=3e-012;           end;
 syms s;
 hie=b (1, 1);        hfe=h (2, 1);        hoe=h (2, 2);
 gm=Ic/26;            hfe=min ( [hfe .95*hie*gm]);
 rbe=hfe/gm;          rbb=hie-rbe;         cbe=gm/ (2*pi*ft);
 Rh=Rb1*Rb2/ (Rb1+Rb2);     RL1=1/ (hoe+1/Rc+1/RL);
 ybe=1/rbe+s*cbe;           zc-1/ (1/rbc+s*Cc);
 A= [1 zc; gm 1] / (1-gm*zc);                          % A矩阵
```

```
A=[1 Rs;0 1]*[1 0;1/Rb 1]*[1 rbb;0 1]*[1 0;ybe 1]*A*[1 0;1/RL1 1];
H=1/A (1, 1);     [b, a] =numden (H);    b=sym2poly (b); a=sym2poly (a);
f=logspace (3, 8, 201);    H=freqs (b, a, 2*pi*f);
Av=20*log10 (abs (H)); Avm=max (Av); I=find(abs(Av-(Avm-3))<.1);
I=fix (mean (I));    f3db=f (I);    Av3db=Av (I);
subplot (211); semilogx (f, Av, [f3db f3db], Avm- [0 20],'r:');      %幅频特性
grid; zoom xon;
xlabel ('frequency (Hz)');   ylabel ('Av (dB)');
tstr= ['f_3_d_B = ' num2str (round (f3db*1e-04) *.01) ' (MHz), '];
tstr= [tstr blanks (6) 'Av_0 = 'num2str (round (Avm*10) *.1) ' dB'];
title (tstr);
subplot (212); semilogx (f, angle (H) *180/pi-180);   grid;          % 相频特性
xlabel ('frequency (Hz)');   ylabel ('Phase (°)');
set (gcf,'units','pix','pos', [200, 120, 560, 420], …
  'name','Common Emitter Amplifer, BBI 2000','num','off');
```

使用缺省参数时的运算结果见图 5.26，晶体管的 h 参数和各个电路元件的参数与图 5.23 中的参数是一样的，因此这个结果是该音频放大器的高频频率响应。图 5.26 中给出了参数的具体数值，增益下降 3dB 时的频率为 710kHz，此时的相移约为 45°。放大器的中频增益为 29.3dB，而图 5.24 中的增益为 35.6dB，注意，后者是指放大器的净增益，即 Av=vo/vi，若计算 Avs=vo/vs，则两者的数值是一样的。

为了提高阻容耦合放大器的上限工作频率，应该采用高频晶体管，高频晶体管的 hie 的数值要小一些，约在 500Ω 左右，因此其基区电阻 $r_{bb'}$ 也比较小，同时其特征频率也要高一些，一般在 400MHz 以上；另外，减小晶体管的工作电流、降低放大器的负载阻抗、降低信号源内阻也可以在一定程度上提高阻容耦合放大器的上限工作频率。例如在视频放大器中，信号源的内阻为 $R_s=75\Omega$，若负载电阻 $R_L=100\Omega$，选用特征频率为 400MHz 的晶体管，而其它参数不变，则 3dB 点的频率提高到 7.5MHz。

5.5.7 共基极放大电路的高频频率响应

分析共基极放大电路的高频频率响应，晶体管应该采用共基极 T 型高频等效电路，这样整个共基极放大器的高频等效电路如图 5.27 所示，其参数可以由混合 π 型高频等效电路的参数和低频 h 参数折算出来。

共基极交流放大系数 $\alpha_0=\beta_0\ (1+\beta_0)\ =h_{fe}/\ (1+h_{fe})$

发射结电阻 $r_e=\alpha_0/g_m$ (Ω)

发射结电容 $C_e=C_{b'e}/\ (1+m)$ (F)

对于均匀基区晶体管 m = 0.2，对于扩散型基区晶体管 m = 0.4。

集电极交流电阻 $r_c=r_{b'c}$ (Ω)

分析共基极放大电路仍可使用 A 矩阵的方法。晶体管共基极等效电路的 A 矩阵可以由以下的方程组求出：

```
» amplif5;

Parameters =
Cb'c            3e-012
Cb'e       1.5303e-010
ft           100000000
gm            0.096154
rbb'               680
rb'e               520
Rb1              24000
Rb2               6000
Rc                2000
RL                6000
»
```

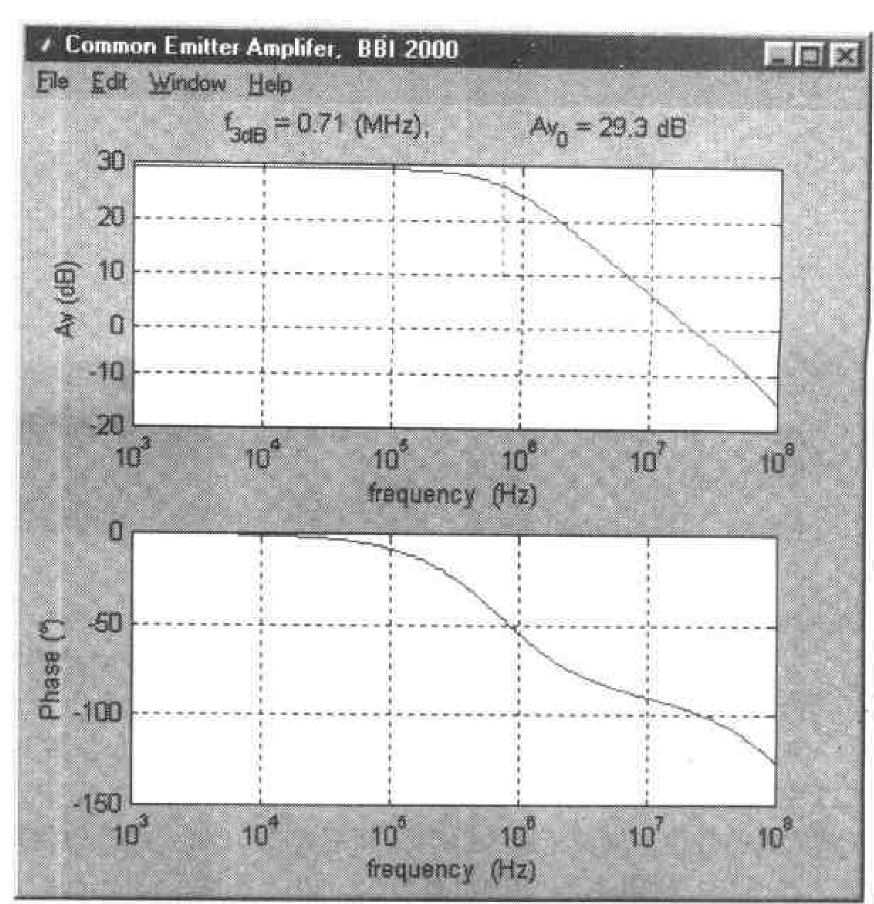


图 5.26 共发射极放大电路的高频频率响应

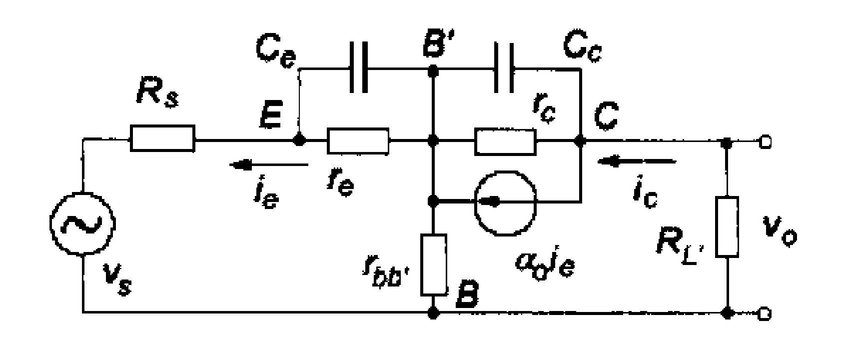


图 5.27 共基极放大器的高频等效电路

$$\begin{cases} i_c - i_e = v_{b'}/r_{bb'} \\ v_e + Z_e \cdot i_e = v_{b'} \\ v_{b'} + Z_c \cdot (i_c - \alpha_0 \cdot i_e) = v_c \end{cases}$$

于是有 $A = \dfrac{1}{r_{bb'} + \alpha_0 Z_c}\begin{bmatrix} r_{bb'} + Z_c & r_{bb'} \cdot [(1-\alpha_0) Z_c + Z_e] + Z_c \cdot Z_e \\ 1 & r_{bb'} + Z_c \end{bmatrix}$

自定义 M 函数 amplif6.m 根据图 5.27 所示的等效电路来分析共基极放大器的高频频率响应，由于使用了符号运算，因此它是工作在 5.x 版环境下的，其用法是：

```
» H=amplif6 (Cc, ft, Ic, RL, h, Rs, rc, m);
```

输入参数：　（其中电阻的单位均为Ω，电容的单位均为F）

Cc　集电极电容，　　ft　晶体管的特征频率

Ic　集电极电流，单位为mA　　RL　负载电阻

Rs　信号源内阻，　　rc　集电极交流电阻

h＝［hie hre；hfe hoe］　晶体管的h参数

m　晶体管基区参数　m＝0.2（均匀基区），0.4（扩散型基区）

输出参数：

H＝vo/vs　放大器的转移函数

amplif6.m的程序清单如下：

```
function [H, Zi] =amplif6 (Cc, ft, Ic, RL, Rs, h, rc, m);
%  Frequency Responce of Common Base Amplifer,    BBI 2000
%
%  Usage: [H, Zi] =amplif6 (Cc, ft, Ic, RL, Rs, h, rc, m);
if nargin<8; m=0.2;          end;      % m=0.2, 0.4
if nargin<7; rc=5e+06;       end;
if nargin<6; h= [1200 3.37e-4; 50 27.1e-6]; end; % [hie hre; hfe hoe]
if nargin<5; Rs=1e+03;       end;
if nargin<4; RL=2e+03;       end;
if nargin<3; Ic=2.5;         end;
if nargin<2; ft=100e+06;     end;
if nargin<1; Cc=3e-012;      end;
syms s;
hie=h (1, 1);        hfe=h (2, 1);        a0=hfe/ (1+hfe);
gm=Ic/26;            hfe=min ( [hfe .95*hie*gm]);
re=a0/gm;            rbb=hie-hfe/gm;
cbe=gm/ (2*pi*ft);        Ce=cbe/ (1+m);
ze=1/ (1/re+s*Ce);        zc=1/ (1/rc+s*Cc);
A= [rbb+ze rbb* ( (1-a0) *zc+ze) +zc*ze; 1 rbb+zc] / (rbb+a0*zc);
A= [1 Rs; 0 1] *A* [1 0; 1/RL 1];
H=1/A (1, 1); [b, a] =numden (H); b=sym2poly (b); a=sym2poly (a);
f=logspace (3, 8, 401);   H=freqs (b, a, 2*pi*f);
Av=20*log10 (abs (H)); Avm=max (Av);
I=find (abs (Av- (Avm-3)) <.1);
I=fix (mean (I));   f3db=f (I);   Av3db=Av (I);
subplot (211); semilogx (f, Av, [f3db f3db], Avm- [0 20],'r:');
grid; zoom xon;
xlabel ('frequency (Hz)');      ylabel ('Av (dB)');
```

```
tstr= ['f_3 d_B = ' num2str (round (f3db * 1e-04)  * .01) ' (MHz), '];
tstr= [tstr blanks (6) 'Av_0 = 'num2str (round (Avm * 10)  * .1) ' dB'];
title (tstr);
subplot (212); semilogx (f, angle (H)  * 180/pi);    rid;
xlabel ('frequency (Hz)');    ylabel ('Phase (°)');
set (gcf, 'units', 'pix', 'pos', [200, 120, 560, 420], ...
   'name', 'Common Base Amplifer, BBI 2000', 'num', 'off');
```

使用缺省参数时的运算结果见图5.28，其中晶体管的h参数与图5.26中的参数是一样的。图5.28中给出了参数的具体数值，增益下降3dB时的频率为13.72MHz，此时的相移约为45°。放大器的中频增益为5.6dB。仿真的结果表明，共基极放大器的增益主要由集电极负载电阻与信号内阻的阻值之比来确定。

从这个仿真结果可以清楚地看出，同一个晶体管工作在共基极组态时的截止频率要高于工作在共发射极组态时的截止频率。

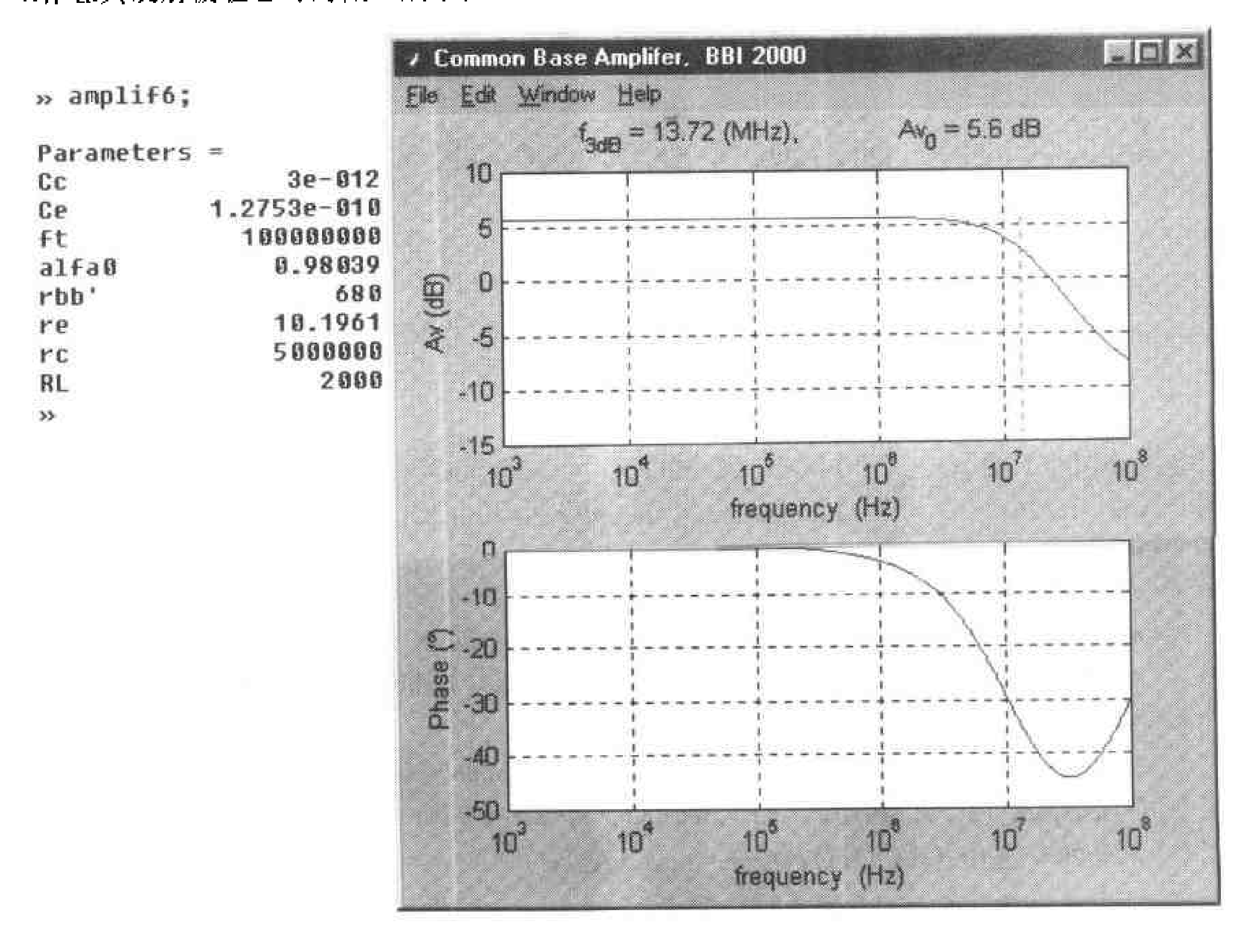


图5.28 共基极放大电路的高频频率响应

5.6 小信号调谐放大器

调谐放大器又称为谐振放大器，其负载通常是具有选频作用的 LC 调谐回路。在小信号调谐放大器中，由于信号电压和电流的幅度都比较小，因此晶体管可以看作是有源线性系统。在工作频率比较高的情况下，晶体管内部的反馈是不能忽略的，这种反馈往往是放大器产生自激的原因，因此在分析调谐放大器时，通常采用晶体管的高频 y 参数等效电路，如图 5.29 所示。

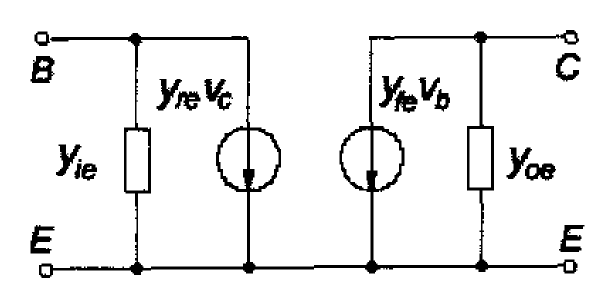


图 5.29　高频 y 参数等效电路

晶体管的高频 y 参数可以测量出来，也可以由混合 π 型参数推出，其表达式为：

$$y_{ie} = \frac{g_{b'e} + j\omega C_{b'e}}{(1 + r_{bb'}g_{b'e}) + j\omega C_{b'e}r_{bb'}}$$

$$y_{re} = \frac{-(g_{b'c} + j\omega C_{b'c})}{(1 + r_{bb'}g_{b'e}) + j\omega C_{b'e}r_{bb'}}$$

$$y_{fe} = \frac{g_m}{(1 + r_{bb'}g_{b'e}) + j\omega C_{b'e}r_{bb'}}$$

$$y_{oe} = g_{ce} + j\omega C_{b'c} + r_{bb'}g_m \cdot \frac{g_{b'c} + j\omega C_{b'c}}{(1 + r_{bb'}g_{b'e}) + j\omega C_{b'e}r_{bb'}}$$

其中 $g_{b'e} = 1/r_{b'c}$，$g_{b'c} = 1/r_{b'e}$，$g_{ce} = 1/r_{ce}$。

以一个 LC 谐振回路作为负载的放大器称为单调谐放大器，图 5.30 为单调谐放大器的交流等效电路，其电压增益和输入导纳的表达式分别为

$$Av(s) = \frac{v_c}{v_b} = \frac{-y_{fe}}{y_{oe} + Y}$$

$$Y_i(s) = y_{ie} - \frac{y_{re}y_{fe}}{y_{oe} + Y}$$

其中，$Y = \frac{1}{R + sL} + sC$。

自定义 M 函数 resonance.m 根据图 5.30 所示的等效电路来分析单调谐放大器的增益和输入导纳，由于使用了符号运算，因此它是工作在 5.x 版环境下的，其用法是：

》[Av, Yi] = resonance (L, C, R, Cc, ft, Ic, h, rbc);

输入参数：　(其中电阻的单位为 Ω，电容的单位均 F，电感的单位为 H)

Cc	集电极电容，	ft	晶体管的特征频率
Ic	集电极电流，单位为 mA	rbc	集电极交流电阻

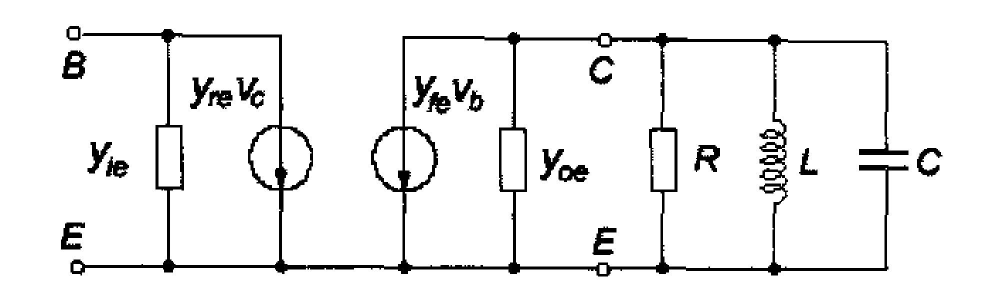


图 5.30 单调谐放大器的交流等效电路

h= [hie hre; hfe hoe]　　晶体管的 h 参数

输出参数:

Av=vc/vb　　放大器的电压增益，　Yi　放大器的输入导纳

resonance.m 的程序清单如下:

```
function [Av, Yi] =resonance (L, C, R, Cc, ft, Ic, h, rhc);
%  Frequency Responce of Resonance Amplifer,    BBI 2000
%
%  Usage: [Av, Yi] =resonance (L, C, R, Cc, ft, Ic, h, rbc);
if nargin<8;  rbc=5e+06;              end;
if nargin<7;  h= [600 3.37e-4; 50 27.1e-6]; end; % [hie hre; hfe hoe]
if nargin<6;  Ic=2;                   end;
if nargin<5;  ft=100e+06;             end;
if nargin<4;  Cc=2e-12;               end;
if nargin<3;  R=100e+03;              end;
if nargin<2;  C=3e-12;                end;
if nargin<1;  L=63e-06;               end;
syms s;
hie=h (1, 1);hfe=h (2, 1);            hoe=h (2, 2);
gm=Ic/26;  bfe=min ( [hfe .95*hie*gm]);
rbe=hfe/gm;  rbb=hie-rbe;             cbe=gm/ (2*pi*ft);
yden= (1+rbb/rbe) +s*cbe*rbb;
yie= (1/rbe+s*cbe) /yden;          yre=- (1/rbc+s*Cc) /yden;
yfe=gm/yden;     yoe=hoe+s*Cc+rbb*gm* (1/rbc+s*Cc) /yden;
Y=1/R+1/ (s*L) +s*C;
Av=-yfe/ (yoe+Y);                  [b, a] =numden (Av);
b=sym2poly (b);                    a=sym2poly (a);
f= (1: .001: 15) *1e+06;           H=freqs (b, a, 2*pi*f);
Av=abs (H);       [Av0, I] =max (Av); f0=f (I); Av=Av/Av0;
```

```
Yi = yie - yre * yfe/ (yoe + Y);           [b, a] = numden (Yi);
b = sym2poly (b);                          a = sym2poly (a);
Yi = freqs (b, a, 2 * pi * f);
I = find (abs (Av - .7071) < .001);        f3 = f (I);
I = find (abs (Av - .1) < .0005);          f20 = f (I);
f202 = max (f20); f201 = min (f20);        f32 = max (f3);   f31 = min (f3);
K01 = (f202 - f201) / (f32 - f31);
f = f * 1e - 06; f0 = f0 * 1e - 06; f201 = f201 * 1e - 06; f202 = f202 * 1e - 06;
f31 = f31 * 1e - 06; f32 = f32 * 1e - 06;
subplot (211); plot (f, Av, [f0 f0], [0 1],'r:', [f201 f202], …
        [.1 .1],'k:', [f31 f32], [.7071 .7071],'k:'); zoom xon;
xlabel ('frequency (MHz)');   ylabel ('Av/Av0');
tstr = ['f_0 = ' num2str (round (f0 * 10) * .1) ' (MHz),   Av_0 = '];
tstr = [tstr num2str (round (Av0)) ',   K_0_._1 = '];
tstr = [tstr num2str (round (K01 * 100) * .01)];      title (tstr);
text (f31, .8, num2str (round (f31 * 10) * .1),'hor','right', …
    'fontname','arial','fontsize', 10);
text (f32, .8, num2str (round (f32 * 10) * .1),'hor','left', …
    'fontname','arial','fontsize', 10);
text (f201, .2, num2str (round (f201 * 10) * .1),'hor','right', …
    'fontname','arial','fontsize', 10);
text (f202, .2, num2str (round (f202 * 10) * .1),'hor','left', …
    'fontname','arial','fontsize', 10);
text (1, .7,'0.7','hor','center','fontname','arial','fontsize', 10);
text (1, .1,'0.1','hor','center','fontname','arial','fontsize', 10);
subplot (212); plot (f, real (Yi), f, imag (Yi)); v = axis; hold on;
grid; zoom xon;   legend ('Gi','Bi');
plot ( [f0 f0], v (3: 4),'r:');    hold off;
xlabel ('frequency (MHz)');   ylabel ('Yi (S)');
set (gcf,'units','pix','pos', [200, 120, 560, 420], …
    'name','Resonance Amplifer, BBI 2000','num','off');
```

使用缺省参数时的运算结果见图 5.31，其中包括了谐振频率、最大增益，矩形系数 ($K_{0.1}=B_{0.1}/B_{0.7071}$，用来表示放大的选择性)，输入电导和输入电纳。特别需要注意的是：在频率低于谐振频率时，输入电导有可能为负值（所谓的负阻），这表示由于内部反馈的存在，晶体管本身实际上是个双向网络，在多级单调谐放大器级联的情况下，负阻现象的出现有可能使谐振放大器处于自激状态。

```
» resonance;

Parameters =
L              6.3e-005
C                3e-012
R                100000
Cc               2e-012
ft            100000000
Ic                    2
gm             0.076923
rbc             5000000
»
```

Resonance Amplifer, BBI 2000

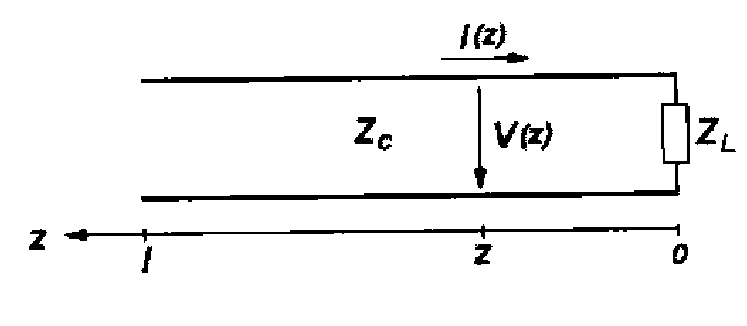
File Edit Window Help

f_0 = 6.5 (MHz), Av_0 = 1148, $K_{0.1}$ = 9.91

图 5.31 单调谐放大器的增益和输入导纳

5.7 传输线

传输线又称为长线，在通信系统和广播系统之中使用的传输线种类是很多的，如：平行双线、同轴电缆、矩形波导、圆形波导、微带线等。传输线是一个分布参数系统，分析传输线的特性要使用长线理论。



5.32 长线的示意图

5.7.1 长线理论

长线的示意图见图 5.32，其上的电压分布和电流分布是求解电报方程而得到的，电报方程的解清楚地表明，长线上电压和电流是由入射波和反射波两部分组成的：

$$\begin{cases} V(z) = V_+(z) + V_-(z) \\ I(z) = I_+(z) - I_-(z) \end{cases}$$

长线上任意一点，电压入射波与电压反射波之比称为该点的电压反射系数，记为 $\Gamma(z)$，而终端的电压反射系数记为 Γ_0，电压反射系数为复数。

$$\Gamma(z) = V(z)/I(z) = \Gamma_0 \exp(-j2\beta z),\ \Gamma_0 = V(o)/I(o) = |\Gamma_0| \exp(-j\varphi_0) = \frac{Z_L - Z_c}{Z_L + Z_c},$$

无耗线的相位常数 $\beta = 2\pi/\lambda$。引入电压反射系数之后，无耗线上的电压、电流分布规律为

$$\begin{cases} V(z) = V_+(z) \cdot [1 + \Gamma(z)] \\ I(z) = I_+(z) \cdot [1 - \Gamma(z)] \end{cases}$$

其中，入射电压 $V_+(z) = \frac{V(0)\exp(j\beta z)}{1+\Gamma_0}$，入射电流 $I_+(z) = \frac{V(0)\exp(j\beta z)}{(1+\Gamma_0) \cdot Z_c}$

长线上任意一点电压与电流之比称为该点的输入阻抗，记为 $Z(z)$，

$$Z(z) = Z_c \frac{1+\Gamma(z)}{1-\Gamma(z)}$$

长线上的电压驻波比为：$VSWR = \frac{V_{max}}{V_{min}} = \frac{1+|\Gamma_0|}{1-|\Gamma_0|}$

无耗线上传输的功率可以表示为：$P = \frac{V_{max}^2}{Z_c \cdot VSWR}$

5.7.2 史密斯圆图

长线中的阻抗和电压反射系数都是复数，工程上通常采用图解法来对阻抗和反射系数进行分析和计算，这样就引出了史密斯圆图。史密斯圆图的实轴代表了反射系数的实部，而其虚轴则代表了反射系数的虚部，因此等 Γ 圆的圆心位于坐标的原点。史密斯圆图的实质就是 z 平面和 Γ 平面两个复平面之间的映射。

史密斯圆图又分为阻抗圆图和导纳圆图两种类型，在采用归一化阻抗和归一化导纳之后，将阻抗圆图旋转 180°就得到了导纳圆图。在阻抗圆图中，等电阻圆的圆心在正的实轴上，而等电抗圆圆心轨迹与虚轴平行，在虚轴的右侧，与虚轴的距离为 1。

设反射系数 $\Gamma = u + jv$，归一化阻抗 $\frac{Z(z)}{Z_c} = r + jx$，是等电阻圆和等电抗圆的方程为：

$$\begin{cases} (u - \frac{r}{1+r})^2 + v^2 = (\frac{1}{1+r})^2 \\ (u-1)^2 + (v - \frac{1}{x})^2 = \frac{1}{x^2} \end{cases}$$

自定义 M 函数 smithch.m 用来对仿真史密斯圆图，其用法之一是：

》A = smithch (zl);

输入参数：　　zl = ZL/Zc 为归一化负载阻抗

输出参数：

A (1) 反射系数模值， A (2) 终端反射系数的幅角，A (3) 驻波比

自定义 M 函数 smithch.m 的用法之二用来对阻抗测量进行仿真，即根据驻波比和波节点 (Vmin) 的位置来计算终端阻抗：

》ZL = smithch ([VSWR zmin Zc]);

输入参数： VSWR 为驻波比， Zc 为特性阻抗

zmin 为电压波节点的位置（单位为波长）

输出参数： ZL 为终端阻抗

smithch.m 的程序清单如下：

```
function [A] =smithch (arg, z, Km, la);
%
%   Usage_1: smithch (zl, z, Km);
%   Usage_2: smithch ( [VSWR zmin Zc]);    BBI 2000
if nargin<4; la=1/8; end;
if nargin<3; Km=0; end;
if nargin<1; arg=1; end;
if length (arg) ==3;
  zl=arg (1); zmin=arg (2); Zc=arg (3); Km=0; ph= (1-zmin*4) *pi;
  gm= (zl-1) / (zl+1) *exp (-j*ph);   z= (1+gm) / (1-gm); Kz=1;
else;
  zl=arg (1); Zc=1; Kz=0;
end;
%------------------------------绘制史密斯圆图
t= (0: 360) *pi/180;   x=cos (t);   y=sin (t);   r= [.99 1.02];
plot (x, y,'k');   hold on;   whitebg ('w');   axis equal; axis off;
H=plot ( [-1.02 1.02], [0 0]);   set (H,'color', [0 .5 0]);
for i=1: 6;
  u=r*cos (pi*i/12);   v=r*sin (pi*i/12);
  plot (u, v,'k', u, -v,'k', -u, v,'k', -u, -v,'k');
end;
text (-1.06, 0,'0','fontname','arial','fontsize', 10, …
      'color', [0 0 .5],'fonta','italic');
text (-.06, 1.04,'0.125','fontname','arial','fontsize', 10, …
      'color', [0 0 .5],'fonta','italic');
text (1.03, 0,'0.25','fontname','arial','fontsize', 10, …
      'color', [0 0 .5],'fonta','italic');
text (-.06, -1.04,'0.375','fontname','arial','fontsize', 10, …
      'color', [0 0 .5],'fonta','italic');
```

```
i=150: 170; xi=x(i)*1.1; yi=y(i)*1.1;
H=plot([-1 xi], [.5 yi],'k', [-1 xi], [-.5 -yi],'k');
text(-1.08, .78,'TO GENERATOR','fontname','arial','fontsize', 9);
text(-1, -.7,'TO LOAD','fontname','arial','fontsize', 9);
plot(.5*x+.5, .5*y,'b');                    % r=1
text(.02, .033,'1','fontname','arial','fontsize', 10,'color','b');
if Km<1;
  i=1: 10: 361;       xi=x(i);       yi=y(i);
  gm=1/3;    H=plot(gm*xi, gm*yi,':');    set(H,'color', [0 .5 0]);
  gm=2/3;    H=plot(gm*xi, gm*yi,':');    set(H,'color', [0 .5 0]);
 text(-.23, .19,'1/3','fontname','arial','fontsize', 10, ...
        'color', [0 .5 0]);
 text(-.58, .48,'2/3','fontname','arial','fontsize', 10, ...
        'color', [0 .5 0]);
  a=1/(1+2); plot(a*x+1-a, a*y,'b');     % r=2
  a=1/(1+.5); plot(a*x+1-a, a*y,'b');    % r=.5
  text(.36, .033,'2','fontname','arial','fontsize', 10,'color','b');
  text(-.42, .033,'0.5','fontname','arial','fontsize', 10,'color','b');
  text(-.97, .033,'0','fontname','arial','fontsize', 10,'color','b');
  xx=x+1; yy=y+1; I=find((xx.*xx+yy.*yy)<=1.001);
  plot(xx(I), yy(I),'r', xx(I), -yy(I),'m');       % x=-1, 1
  a=1/2; xx=a*x+1; yy=a*y+a; I=find((xx.*xx+yy.*yy)<=1.001);
  plot(xx(I), yy(I),'r', xx(I), -yy(I),'m');       % x=-2, 2
  a=1/.5; xx=a*x+1; yy=a*y+a; I=find((xx.*xx+yy.*yy)<=1.001);
  plot(xx(I), yy(I),'r', xx(I), -yy(I),'m');          % x=-.5, .5
  text(.53, .79,'2','fontname','arial','fontsize', 10,'color','r');
  text(-.04, .95,'1','fontname','arial','fontsize', 10,'color','r');
  text(-.63, .72,'1/2','fontname','arial','fontsize', 10,'color','r');
  text(.52, -.79,'-2','fontname','arial','fontsize', 10,'color','m');
  text(-.05, -.95,'-1','fontname','arial','fontsize', 10,'color','m');
  text(-.65, -.72,'-1/2','fontname','arial','fontsize', 10,'color','m');
end;
%-----------------------------------
gm=(zl-1)/(zl+1);    g1=real(gm);    g2=imag(gm);
ph0=round(angle(gm)*1800/pi)*.1;    gm=abs(gm);
plot(gm*x, gm*y,'k');       if (Km+Kz)==0; plot(g1, g2,'k*'); end;
VSWR=(1+gm)/(1-gm-(gm==1)*eps);       A=[gm ph0 VSWR];
if gm>0.1;
  ph=190*pi/180; u=1.01*gm*cos(ph); v=1.01*gm*sin(ph);
```

```
text (u, v, ['G = ' num2str (round (gm * 100) * .01)],'fontname','symbol', …
  'fontsize', 10,'hor','right');
text (-1, -1, ['VSWR = ' num2str (round (VSWR * 100) * .01)], …
    'fontname','arial','fontsize', 10);
if Km= =1;                                    %单支节匹配
  u=gm * gm; v=sqrt (.25- (u-.5) ^2); x1=2 * v/ ( (1-u) ^2+v * v);
  a=1/x1; xx=a * x+1; yy=a * y+a; I=find ( (xx. * xx+yy. * yy) <=1.002);
  plot (-g1, -g2,'k * '); al=atan2 (-g2, -g1) +pi; zmax=al * .25/pi;
  text (-1.2 * g1, -1.2 * g2,'Y_L','fontname','arial', …
    'fontsize', 10,'hor','center');
  H=plot ( [0 -g1], [0 -g2],':'); set (H,'color', [0 .6 0]);
  H=plot (xx (I), yy (I),'r', xx (I), -yy (I),'m');       % b
  u1=u/gm; v1=v/gm; plot ( [0 u1], [0 v1],'k:', [0 u1], [0 -v1],'k:');
  x=gm * gm; y=sqrt (x-x * x); al=atan (y/x);
  d1=(1-al/pi)*.25; d1=round(d1*1000)*.001; u2=1.13*u1; v2=1.13*v1;
  text (u2, v2, ['d1 = ' num2str (d1)],'fontname','arial', …
    'fontsize', 9,'fonta','italic','color', …
     [.8 0 .8],'hor','center');
  text (u2, -v2, ['d2 = ' num2str (.5-d1)],'fontname','arial', …
    'fontsize', 9,'fonta','italic','color','r','hor','center');
  text (u2+.17, v2,'l','fontname','symbol','color', …
     [.8 0 .8],'fontsize', 10);
  text (u2+.17, -v2,'l','fontname','symbol','color','r', …
    'fontsize', 10);
  v=2/x1/ (1+1/x1/x1);         u=1-v/x1;
  al=atan2 (v, u);             al=al+ (al<0) * pi;
  u2=1.15 * cos (al);          v2=1.15 * sin (al);
  L1=.25 * al/pi;              L1=round (L1 * 1000) * .001;
  text (u2, -v2, ['L1 = ' num2str (L1)],'fontname','arial', …
    'fontsize', 9,'fonta','italic','color', …
     [.8 0 .8],'hor','center');
  text (u2, v2, ['L2 = ' num2str (.5-L1)],'fontname','arial', …
    'fontsize', 9, 'fonta','italic','color', …
    'r','hor','center');
  text (u2+.17, -v2,'l','fontname','symbol', …
    'color', [.8 0 .8],'fontsize', 10);
  text (u2+.17, v2,'l','fontname','symbol','color','r','fontsize', 10);
  A= [d1 .5-d1 zmax; L1 .5-L1 zmax];
elseif Km= =2;               %双支节匹配, d=1/8 lamhda
```

```
u1=0.5*x; v1=0.5*y+0.5;
H=plot (u1, v1); set (H,'color', [0 .7 0]);
gm=(zl-1)/(zl+1); g1=real(-gm); g2=imag(-gm); plot(g1,g2,'k*');
text (1.1*g1, 1.1*g2,'Y_L','fontname','arial', ...
  'fontsize', 10,'hor','center');
gm=gm*exp (-j*4*pi*la);    y1= (1-gm) / (1+gm);
g=real (y1);          ba1=imag (y1);
if g>2;
  gm=gm*exp (-j*pi/2); y1= (1-gm) / (1+gm);
  g=real (y1); ba1=imag (y1); Km=3;
end;   P=str2mat ('AB','BC');
g1=real (-gm); g2=imag (-gm); plot (g1, g2,'b*');
text (1.2*g1, 1.2*g2, [P (Km-1, 1) ''''],'fontname','arial', ...
  'fontsize', 10,'color','b','hor','center');
a=1/abs (ba1); xx=a*x+1; yy=a*y+a;
I=find ( (xx.*xx+yy.*yy) <=1.01);
c='mr';   plot (xx (I), sign (ba1) *yy (I), c (.5*sign (ba1) +1.5));
a=1/ (1+g); u2=a*x+1-a; v2=a*y; plot (u2, v2,'b');          % A': g
if ahs (g-1) <.001;
  u=g1; v=g2;    uu=u; vv=-v;
else;
  t= (0: .001: .705) *pi; u=.5*sin (t); v=.5-.5*cos (t);
  I=find (ahs ( (u-g*a) .^2+v.*v-a*a) <.001); m=fix (mean (I));
  u=u (m); v=v (m); uu=v; vv=-u;
  end;
  plot (u, v,'r*', uu, vv,'m*');
  text (1.2*u, 1.2*v, P (Km-1, 1),'fontname','arial','fontsize', 10, ...
    'color','r','hor','center');
text (1.2*uu, 1.2*vv, P (Km-1, 2),'fontname','arial','fontsize', 10, ...
    'color','m','hor','center');
  gm=- (u+j*v); y1= (1-gm) / (1+gm); g=real (y1); ba=imag (y1);
  a=1/abs (ba); xx=a*x+1; yy=a*y+a;
  I=find ( (xx.*xx+yy.*yy) <=1.01);
  plot (xx (I), sign (ba) *yy (I), c (.5*sign (ba) +1.5));          % A: b
  gm=- (v-j*u); y1= (1-gm) / (1+gm); g=real (y1); bb=imag (y1);
  a=1/abs (hh); xx=a*x+1; yy=a*y+a;
  I=find ( (xx.*xx+yy.*yy) <=1.01);
  plot (xx (I), sign (bb) *yy (I), c (.5*sign (bb) +1.5));          % B: b
  ph0=atan (v/u); ph= (0: -10: -90) *pi/180+ph0; rh=sqrt (u*u+v*v);
```

```
  H=plot (rh * cos (ph), rh * sin (ph),':'); set (H,'color', [0 .6 0]);
  b1=ba-ba1; b1=round (b1 * 1000) * .001; b2= -round (bb * 1000) * .001;
  if Km= =2;
    bstr=str2mat ( ['b1: ' num2str (b1)], ['b2: ' num2str (b2)]);
    A= [b1 b2];
elseif Km= =3;
    bstr=str2mat ('b1: Inf', ['b2: ' num2str (b1)], ...
        ['b3: ' num2str (b2)]);
    A= [inf b1 b2];
end;
text (.94, .9, bstr,'fontname','arial','fontsize', 10, ...
    'color', [0 0 .6]);
  end;
end;
% ------------------------------------
if exist ('z') & Km<1;
  text (.8, 1, ['z1 = ' num2str (round (z * 100) * .01)],
      'fontname','arial','fontsize', 10);
  text (.8, .9, ['y1 = ' num2str (round (1/z * 100) * .01)],'fontname',
      'arial','fontsize', 10);
  gm= (z-1) / (z+1);    g1=real (gm); g2=imag (gm);
  gm=abs (gm); ph=atan2 (g2, g1) * 180/pi; ph=round (ph * 10) * .1;
  plot ( [0 g1/gm], [0 g2/gm],'k:', g1, g2,'k * ');
  al=atan2 (g2, g1);          al=round (al * 180/pi);
  L= (1-al/180) /4;          L=round (L * 1000) * .001;
  text (.8, .8, ['L = ' num2str (L) ' ( ' num2str (al) '° )'], ...
    'fontname','arial','fontsize', 10);
  A= [z 1/z gm ph];
end;
if Kz= =1;
  nstr= ['VSWR = ' num2str (VSWR) ' , zmin = ' num2str (zmin)];
  nstr= [nstr ' lambda , Z = ' num2str (round (z * Zc * 10) * .1) ' Ohm'];
  A=z * Zc;
elseif Km= =1;
  nstr='Smith Chart:          Stub Matching, BBI 2000';
elseif Km= =2;
  nstr='Smith Chart:          Two Stubs Matching, BBI 2000';
elseif Km= =3;
  nstr='Smith Chart:          Three Stubs Matching, BBI 2000';
```

```
else;
  nstr='Smith Chart ,            BBI 2000';
end;
set (gcf,'name', nstr,'num','off','color','w'); hold off;
```

例 1 负载阻抗 $Z_L=100+50j$，特性阻抗 $Z_C=75$，求驻波比和反射系数。

键入 A = smithch ((100+50j) /75)，便得到图 5.33 所示的结果。

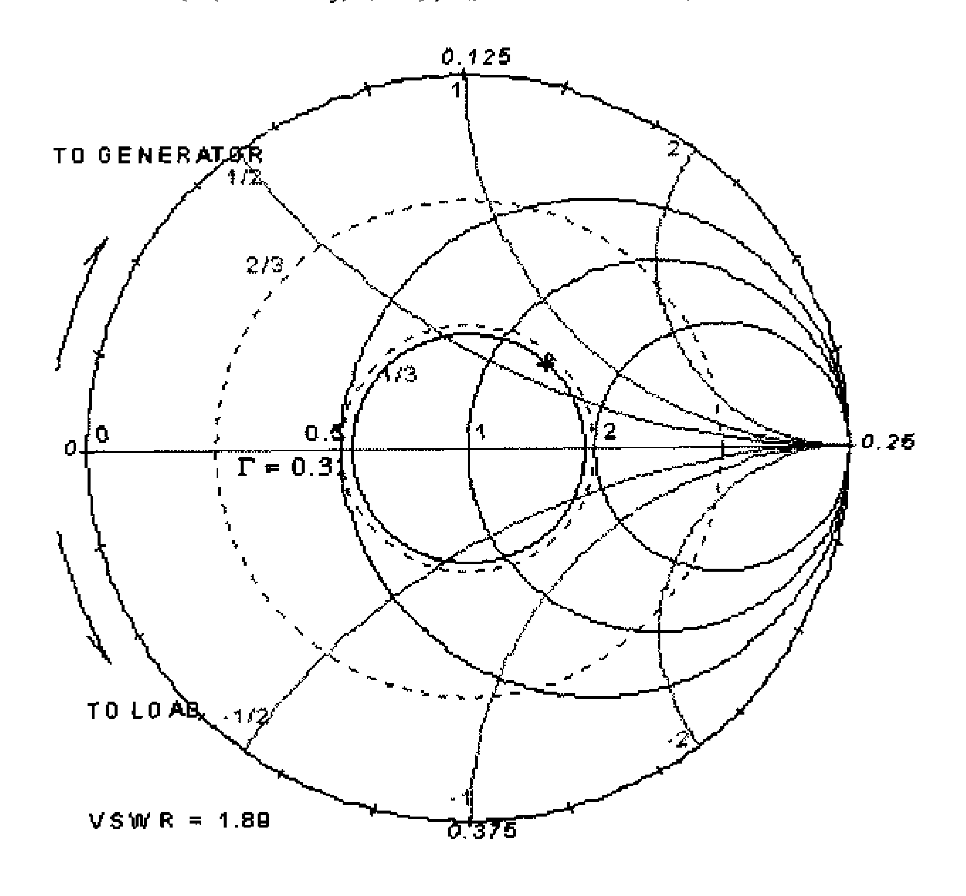


图 5.33 史密斯阻抗圆图的应用之一

例 2 传输线上的驻波比 VSWR=2，电压波节点的位置 $z = 0.2\lambda$，特性阻抗 $Z_C=75$，求负载阻抗。

键入 A= smithch ([2 0.2 75])，便得到图 5.34 所示的结果，图中右上角给出的数值是归一化阻抗和归一化导纳。在圆图的波节线上沿着等 Γ 圆向负载方向转动 0.2λ，于是负载阻抗为 $Z_L=116.6-j51.4$。

5.7.3 阻抗匹配器

在传输线经常使用单支节、双支节或三支节阻抗匹配器，如图 5.35 所示，使用史密斯圆图可以清楚地表明这些阻抗匹配器的工作原理，并计算出其参数。

图 5.35a 是使用短截线构成的单支节匹配器，它经常出现在短波天线的馈电系统中用来进行阻抗匹配，其位置和短截线的长度是可以分别调整的，调整短截线的长度相当于调节电

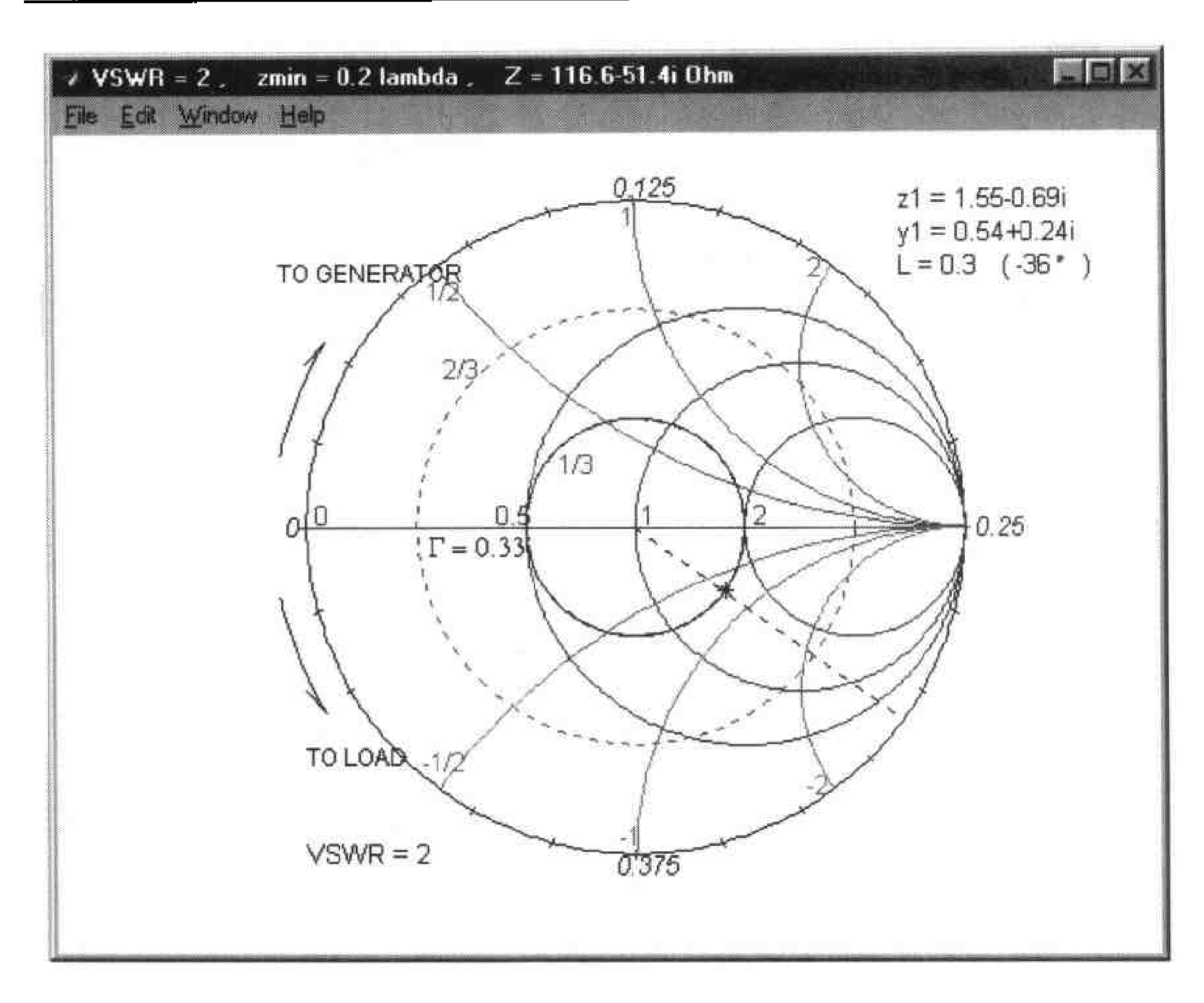


图 5.34 史密斯阻抗圆图的应用之二

抗。

图 5.35b 是双支节匹配器，它通常应用在同轴系统或波导系统中，可调电抗可以用螺钉或活塞来完成，而其位置则是固定的。在双支节匹配器中存在着一定的“匹配盲区”，一旦阻抗进入该区内，阻抗调节是无效的。为了弥补双支节匹配器存在匹配盲区的不足，可以采用图 5.35c 所示的三支节匹配器，将阻抗转移出匹配盲区。

使用 smithch.m 函数可以分析单支节、双支节和三支节阻抗匹配器，其用法是：

》A = smithch (zl, z, Km)；

输入参数：

zl = Z_L/Z_c 归一化负载阻抗，z 阻抗（在分析阻抗匹配器时可以给任意值）

K_m 匹配参数，$K_m=0$ 无匹配、$K_m=1$ 单支节匹配、$K_m=2$ 双支节匹配，若进入匹配盲区，则自动改为三支节匹配。

输出参数：　单支节　A= [d1 d2 zmax; L1 L2 zmax]；

　　　　　　双支节　A= [b1 b2]；　　三支节 A= [b1 b2 b3]

例 3　传输线的负载阻抗为 $Z_L=100+j100$，特性阻抗 $Z_C=75$，单支节匹配。

键入 A = smithch ((100+100j) /75, 1, 1)，便得到图 5.36 所示的结果。电压波腹的

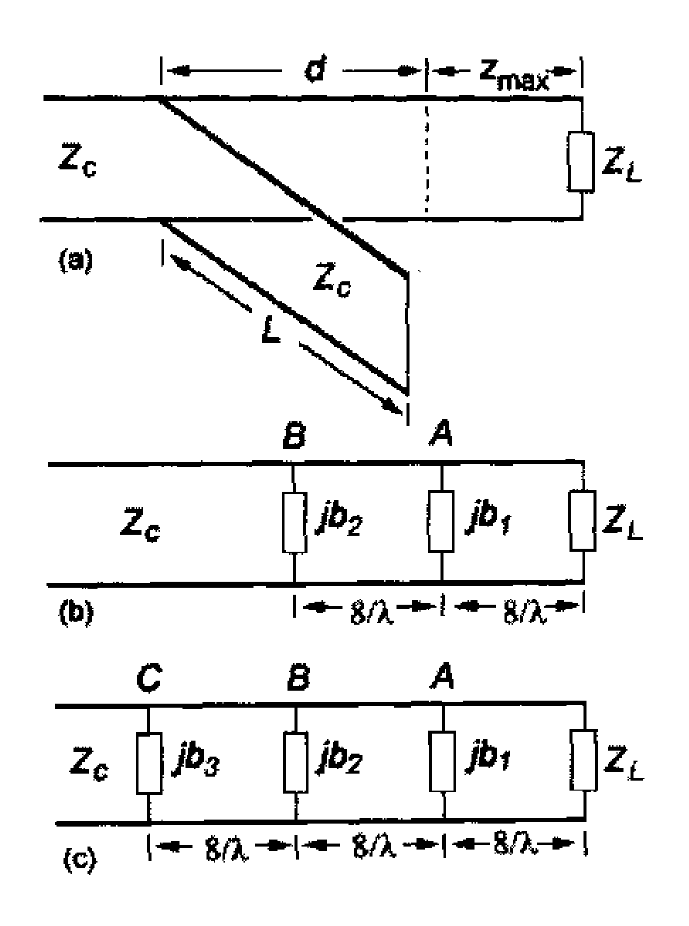


图 5.35　三种阻抗匹配器

位置为 zmax，单支节匹配器的数据有两组：L1 和 d1、L2 和 d2，它们都标注在图形的右上方，其中 d 为短截线与电压波腹（波峰）点的距离，L 为短截线的长度。

例 4　传输线的负载阻抗为 $Z_L = 200 - j100$，特性阻抗 $Z_C = 75$，双支节匹配。

键入 A = smithch（（200 - 100j）/75，1，2），便得到图 5.37 所示的结果，其中两个电纳的数值均标注在图形的右上角处，而双支节匹配器的结构如图 5.35（b）所示。负载导纳在圆图中沿等 Γ 圆转到 A′处，并联上电纳 jb1 变化到 A 处，然后沿等 Γ 圆转到 B 处，最后并联上电纳 jb2，就完全匹配了。

例 5　传输线的负载阻抗为 $Z_L = 45 - j60$，特性阻抗 $Z_C = 75$，三支节匹配。

键入 A = smithch（（45 - 60j）/75，1，2），由于负载导纳转动 90°（λ/8）进入了“匹配盲区”，因此程序自动转变为三支节匹配，于是便得到图 5.38 所示的结果，其中三个电纳的数值均标注在图形的右上角。

5.7.4　分析传输线问题的 M 函数 trans.m

自定义的 M 函数 trans.m 专门用来分析传输线问题，它采用了按钮式界面，可以灵活地输入各种参数，使用方便；同时它还可以显示传输线上的电压分布、电流分布和阻抗分布的情况，计算阻抗匹配器的参数，使用鼠标还可在传输线或史密斯圆图上随意选择感兴趣的点。该函数对传输线的仿真是比较全面的。

trans.m 函数要调用 trans1.m 函数和 smithch.m 函数，其用法是：》trans；

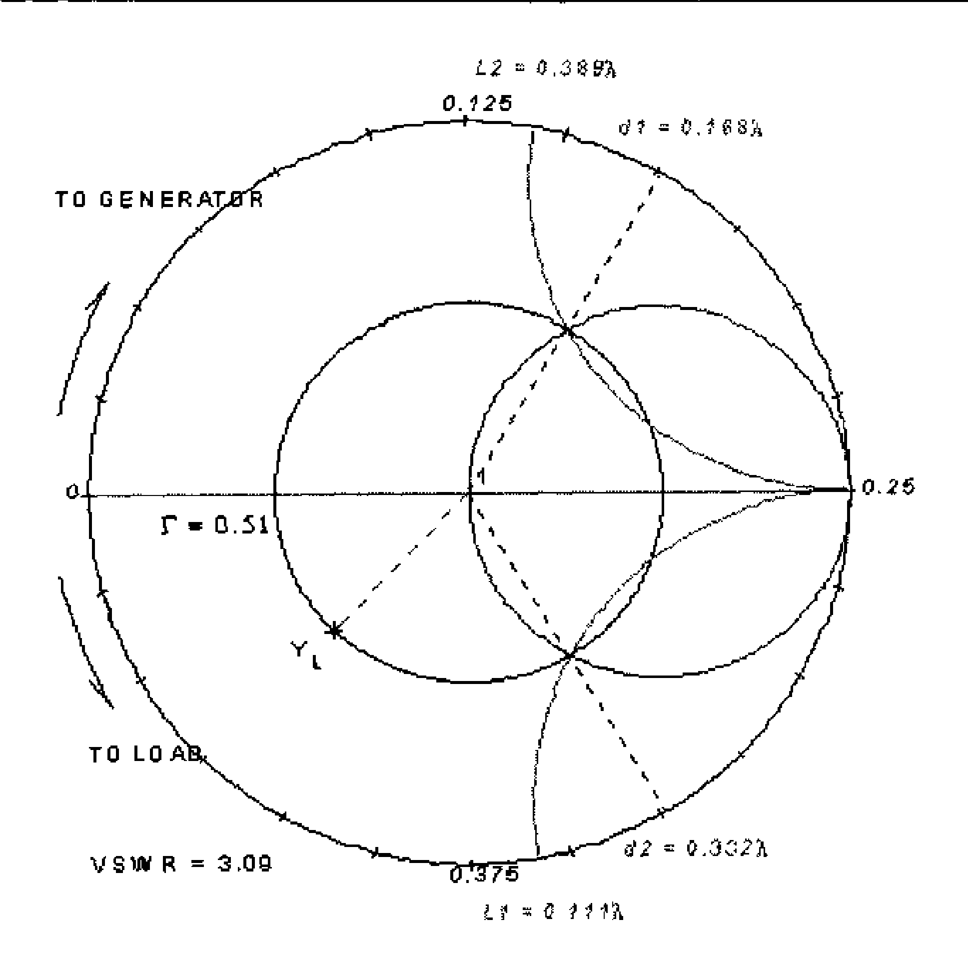


图 5.36 单支节匹配

trans.m 的程序清单如下：

```
function [] =trans (action);
global Zc ZL Vi z z1 H2 H31 H32 H51 H52 H53 H54 H55 H56…
      H57 H58 H59 H62 H71 H72 H73 H74
if nargin<1;
  action='initialized'; H21=-1; H31=-1; H41=-1; H51=-1; H61=-1;
end;
if strcmp (action,'initialized');
  figure ('Name','TRANSMISSION LINE, BBI 2000', …
         'NumberTitle', 'off','Units','pix', …
         'pos', [5 29 792 530],'Color', [1 1 1]);
  axes ('Units','Normal','Position', [0.08 0.1 .7 .84],'vis','off');
  H0=uicontrol ('Style','Frame','Units','Normal','Position', …
                [.82 0 .2 1],'Back', [.8 .8 .8]);
  H1=uicontrol ('Style','Popup','Units','Normal','Position', …
         [.83 .94 .16 .04],'String', str2mat ('Zc= 75, Yc=13.3 mS', …
```

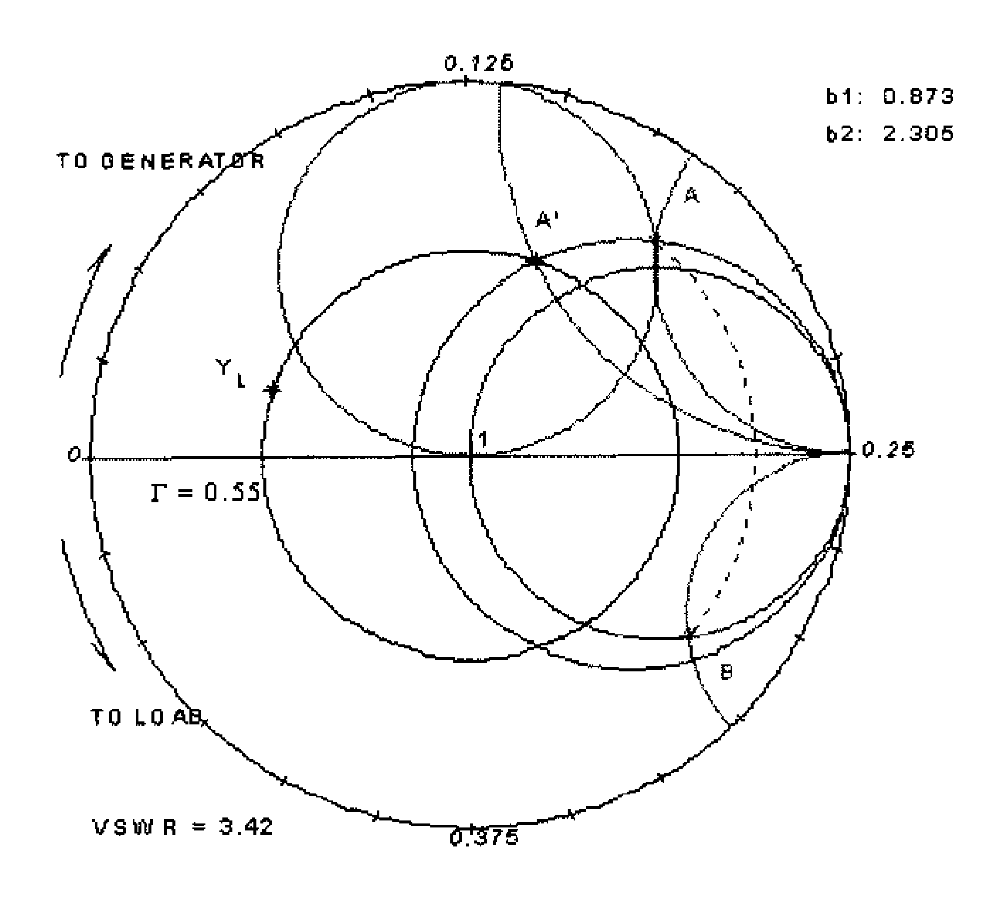


图 5.37 双支节匹配

```
    'Zc= 50, Yc= 20 mS','Zc=100, Yc=10 mS', …
    'Zc=300, Yc= 3.3 mS','Zc=600, Yc= 1.7 mS'), …
    'Back', [1 1 1],'Callback','trans Zc;');
H20=uicontrol ('Style','Text','Units','Normal','Position', …
    [.822 .87 .06 .03],'String','ZL (Ohm)','Back', [.8 .8 .8]);
H2=uicontrol ('Style','Edit','Units','Normal','Position', …
    [.88 .87 .11 .04],'Back', [1 1 1],'String','150', …
    'Callback','trans ZL;');
H21=uicontrol ('Style','Text','Units','Normal','Position', …
    [.83 .8 .06 .03],'String','Vi (V)','Back', [.8 .8 .8]);
H22=uicontrol ('Style','Edit','Units','Normal','Position', …
    [.88 .8 .11 .04],'Back', [1 1 1],'String','1', …
    'Callback','trans Vi;');
H31=uicontrol ('Style','Radio','Units','Normal','Position', …
    [.84 .73 .14 .04],'String','V/A, Impedance','Back', …
    [.8 .8 .8],'value', 1,'Callback','trans curve;');
H32=uicontrol ('Style','Radio','Units','Normal','Position', …
    [.84 .68 .14 .04],'String','Smith Chart','Back', …
```

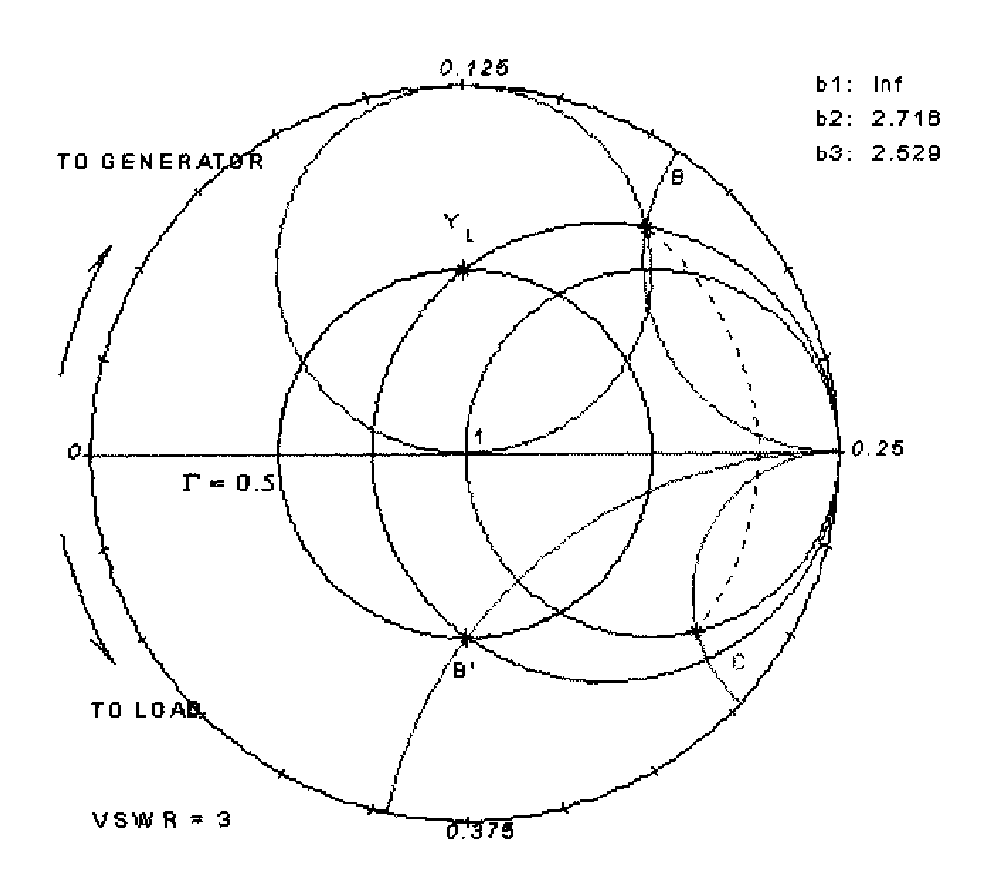


图 5.38 三支节匹配

```
        [.8 .8 .8],'value', 0,'Callback','trans smith;');
H4=uicontrol ('Style','Push','Units','Normal','Position', …
        [.84 .63 .14 .04],'String','Point on Smith Chart', …
        'Callback','trans point;');
H51=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .58 .16 .03],'String','Gamma0: 0.333, 0°', …
        'Back', [1 1 1]);
H52=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .54 .16 .03],'String','VSWR: 2, K: 0.5', …
        'Back', [1 1 1]);
H53=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .50 .16 .03],'String','Return Loss = 9.5 dB', …
        'Back', [1 1 1]);
H54=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .46 .16 .03],'String','Vmax = 1 V,','Back', [1 1 1]);
H55=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .42 .16 .03],'String','Pmax = 10 mW,','Back', [1 1 1]);
H61=uicontrol ('Style','Push','Units','Normal','Position', …
```

```
        [.825 .37 .11 .04],'String','z in wavelength:', …
        'call','trans choose;');
  H62=uicontrol ('Style','Edit','Units','Normal','Position', …
        [.94 .37 .05 .04],'Back', [1 1 1],'String','1', …
        'Callback','trans z;');
  H71=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .32 .16 .03],'String','V (z) = 1V, 0°','Back', [1 1 1]);
  H72=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .28 .16 .03],'String','I (z) = 1V, 0°','Back', [1 1 1]);
  H73=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .24 .16 .03],'String','Z (z) = 1V, 0°','Back', [1 1 1]);
  H74=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .2 .16 .03],'String','Y (z) = 1V, 0°','Back', [1 1 1]);
  H81=uicontrol ('Style','Push','Units','Normal','Position', …
        [.84 .15 .14 .04],'String','One-Stub Match', …
        'Callback','trans stub1;');
  H82=uicontrol ('Style','Push','Units','Normal','Position', …
        [.84 .10 .14 .04],'String','Two-Stubs Match', …
        'Callback','trans stub2;');
  H90=uicontrol ('Style','Text','Units','Normal','Position', …
        [.83 .06 .16 .03],'String', …
        'Rotating on Smith Chart','Back', [1 1 1]);
  H9=uicontrol ('Style','Slider','Units','Normal','Position', …
        [.83 .02 .16 .03],'slider', [1/72 .25],'Value', 0, …
        'Call','trans slider;');
  uimenu ('Label','&Match','call','trans match;')
%------------------------------------------
ZL=150; Zc=75; Vi=1; z=1; trans1 (ZL, Zc);
elseif strcmp (action,'Zc');
  k=get (gco,'Value'); Z= [75 50 100 300 600]; Zc=Z (k);
  trans1 (ZL, Zc); set (H32,'value', 0); set (H31,'value', 1);
elseif strcmp (action,'ZL');
  k=get (gco,'string'); ZL=str2num (k);
  trans1 (ZL, Zc); set (H32,'value', 0);   set (H31,'value', 1);
elseif strcmp (action,'Vi');
  k=get (gco,'string'); Vi=str2num (k);
elseif strcmp (action,'z');
  k=get (gco,'string'); z=str2num (k);
  trans1 (ZL, Zc, z);      set (H32,'value', 0);      set (H31,'value', 1);
```

```
elseif strcmp (action,'choose');
    [x, y] =ginput (1); z=round ( (1-x/10)  *1000) * .001; transl (ZL, Zc, z);
  set (H62,'string', num2str (z));       set (H32,'value', 0);
  set (H31,'value', 1);
elseif strcmp (action,'curve');
  set (H32,'value', 0);    transl (ZL, Zc, z);
elseif strcmp (action,'smith');
  set (H31,'value', 0);    smithch (ZL/Zc, z1);
elseif strcmp (action,'slider');
  if get (H32,'value');
    k=get (gco,'value');     k=round (k*72) /72;
    gm= (ZL-Zc) / (ZL+Zc)  *exp (-j*2*pi*k); z1= (1+gm) / (1-gm);
    smithch (ZL/Zc, z1);
  end;
elseif strcmp (action,'stub1');
  kv=get (H31,'value');
  if kv==1;
    transl (ZL, Zc, z, 1);
  else;
    smithch (ZL/Zc, z1, 1);
  end;
elseif strcmp (action,'stub2');
  smithch (ZL/Zc, z1, 2);      set (H31,'value', 0);       set (H32,'value', 1);
elseif strcmp (action,'point');
  k=get (H32,'value');
  if k==1;
     [x, y] =ginput (1); z1= (1+x+y*j) / (1-x-y*j);
     smithch (ZL/Zc, z1);
  end;
elseif strcmp (action,'match');
  figure ('name','Stub Matching','num','off','pos', ...
              [20 100 300 400],'color','w');
  X=imread ('stub.jpg');    image (X); axis off;
  set (gca,'units','pix','pos', [2 10 288 384]);
end;
% ==================================
gamma0= (ZL-Zc) / (ZL+Zc); g1=abs (gamma0);
g2=angle (gamma0); g2=round (g2*1800/pi)  * .1;
if g1==1; VSWR=inf; else; VSWR= (1+g1) / (1-g1); end;
```

```
L=round (-200*log10 (g1+ (g1==0) *eps)) *.1;
Vmax=Vi* (1+g1);                    Pmax=Vmax^2/ (Zc*VSWR);
Vmax=round (Vmax*100) *.01;         Pmax=round (Pmax*1000) *.001;
if VSWR>1000;            VSWR=inf; K=0;
elseif VSWR==1;     K=1;    L=inf;
else;                    K=round (100/VSWR) *.01;
end;
set (H51,'string', ['Gamma0 = ' num2str (round (g1*1000) *.001) ...
        ', ' num2str (g2) '°']);
set (H52,'string', ['VSWR = ' num2str (round (VSWR*100) *.01) ...
        ', K = ' num2str (K)]);
set (H53,'string', ['Return Loss = ' num2str (L) ' dB']);
set (H54,'string', ['Vmax = ' num2str (Vmax) ' V']);
set (H55,'string', ['Pmax = ' num2str (Pmax) ' W']);
V=Vi* (exp (j*2*pi*z) +gamma0*exp (-j*2*pi*z));
I=Vi/Zc* (exp (j*2*pi*z) -gamma0*exp (-j*2*pi*z));
if abs (I) <1e-04*Vi/Zc; Z=inf; Y=0;
elseif abs (V) <1e-04*Vi; Z=0; Y=inf;
else;      Z=V/I; Y=I/V; Z=round (Z*10) *.1; Y=round (Y*10000) *.1;
end;       z1=Z/Zc;
V1=round (abs (V) *10) *.1;              V2=round (angle (V) *180/pi);
I1=round (abs (I) *10000) *.1;           I2=round (angle (I) *180/pi);
set (H71,'string', ['V (z) = ' num2str (V1) ' V, ' num2str (V2) '°']);
set (H72,'string', ['I (z) = ' num2str (I1) ' mA, ' num2str (I2) '°']);
set (H73,'string', ['Z (z) = ' num2str (Z) ' Ohm']);
set (H74,'string', ['Y (z) = ' num2str (Y) ' mS']);
```

---

trans1.m 函数的程序清单如下：

```
function [] =trans1 (ZL, Zc, z0, Km);
%
% BBI 2000
if nargin<4; Km=0;          end;
if nargin<3; z0=1;          end;
if nargin<2; Zc=75;         end;
if nargin<1; ZL=300;        end;
H=plot ( [0 10], [0 0] -.1,'k', [0 10], [1 1] -.1,'k'); set (H,'linewidth', 2);
axis ( [-1 11 -.1 10]); axis off; bold on;%
plot ( [10 10], [0 1] -.1,'k');
fill ( [9.9 10.1 10.1 9.9], [.22 .22 .78 .78] -.1,'w');
```

```
text (10.3,  .4,'ZL','fontname','arial','fontsize', 10,'fonta','italic');
text (5.4,  .4,'Zc','fontname','arial','fontsize', 10,'fonta','italic');
text (-.4,  .4,'Vi','fontname','arial','fontsize', 10,'fonta','italic');
plot ( [-.5 10], [1.5 1.5],'k', 0, 1.5,'k+', 5, 1.5,'k+', 2.5, 1.5,'k+');
plot (7.5, 1.5,'k+', [10 10], [1.5 3.5],'k', 10, 3.5,'k+');
plot ( [-.5 10], [2.5 2.5],'k:', 10, 2.5,'k+');
text (-.55, 1.5,'<');
text (-.8, 1.5,'z','fontname','arial','fontsize', 10,'fonta','italic');
text (10, 1.25,'0','fontname','arial','fontsize', 9,'fonta', …
        'italic','hor','center');
text (0, 1.25,'l','fontname','symbol','fontsize', 10,'hor','center');
text (2.5, 1.25,'3l/4','fontname','symbol','fontsize', 10, …
        'hor','center');
text (5, 1.25,'l/2','fontname','symbol','fontsize', 10,'hor','center');
text (7.5, 1.25,'l/4','fontname','symbol','fontsize', 10, …
        'hor','center');
text (10.2, 1.5,'0','fontname','arial','fontsize', 9,'fonta','italic');
text (10.2, 2.5,'1','fontname','arial','fontsize', 9,'fonta','italic');
text (10.2, 3.5,'2','fontname','arial','fontsize', 9,'fonta','italic');
%-----------------------------------------------------------------
z=0: .001: 1;          zz=10 * (1-z);        gamma0= (ZL-Zc) / (ZL+Zc);
gamma=abs (gamma0) * exp ( (angle (gamma0) -4 * pi * z) * 1j);
V=ahs (1+gamma); [Vmax, i1] =max (V); [Vmin, i2] =min (V);
plot (zz, V+1.5,'b');
if i1>500; i1=i1-500; end; if i2>500; i2=i2-500; end;
text (zz (i1), Vmax+1.5,'Vmax','fontname','arial','fontsize', 10, …
   'hor','center','ver','bottom','color','b');
text (zz (i2), Vmin+1.5,'Vmin','fontname','arial','fontsize', 10, …
   'hor','center','ver','top','color','b');
if Km<1;
  zmin=10 * (1-z (i2)); plot ( [0 0] +zmin, [-.1 1],'b:');
end;   imax=i1;
I=abs (1-gamma); [Imax, i1] =max (I); [Imin, i2] =min (I);
H=plot (zz, I+1.5); set (H,'color', [0 .5 0]);
if i1<500; i1=i1+500; end; if i2<500; i2=i2+500; end;
text (zz (i1), Imax+1.5,'Imax','fontname','arial','fontsize', 10, …
   'hor','center','ver','bottom','color', [0 .5 0]);
text (zz (i2), Imin+1.5,'Imin','fontname','arial','fontsize', 10, …
   'hor','center','ver','top','color', [0 .5 0]);
```

```
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
if abs (gamma0) >.999;
   ph=angle (gamma0) -4*pi*z; ph=ph+ (ph==0) *eps;
   Z=j*Zc*cot (ph/2); K=0;
else;
   Z=Zc* (1+gamma) ./ (1-gamma); K=1;
end; Y=1./Z;
plot ( [-.5 10], [7 7],'k', 0, 7,'k+', 5, 7,'k+', 2.5, 7,'k+', 7.5, 7,'k+');
plot ( [10 10], [9.5 4.5],'k');     text (-.55, 7,'<');
text (-.8, 7,'z','fontname','arial','fontsize', 10,'fonta','italic');
text (10, 9.7,'R, X','fontname','arial','fontsize', 10,'hor','center');
X=imag (Z);    I=find (abs (X) <=2000);
X=X (I);    Xm=max (abs (X));     zz1=zz (I);
R=real (Z); [Rm, i] =max (R); Zm=max (Xm, Rm); X=X/Zm; R=R/Zm;
plot (zz, 2.5*R+7); if i<500;     i=i+500; end;
if Rm>1;
  text (zz (i), 2.5+7, num2str (round (Rm)),'fontname','arial','fonta', …'italic','
  fontsize', 10,'hor','center','ver','bottom','color','b');
end;
J=find (diff (I) >1); m=length (J) +1; J= [J length (I)];    k=1;
for i=1: m;
   kn=k: J (i); k=J (i) +1;
   H=plot (zz1 (kn), 2.5*X (kn) +7); set (H,'color', [0 .5 0]);
end;
if z0<1;
   z0=10* (1-z0);
   H=plot ([z0 z0], [-.2 1],':', [z0 z0], [1.5 3.5],':', [z0 z0], [4.5 9.5],':');
   set (H,'color', [0 0 .6]);
end;
% ====================================
if Km>0 & abs (gamma0) <.99 & abs (gamma0) >.1;            %Match Point
   Yc=1/Zc;    I=find (abs (real (Y (1: 500)) -Yc) <1e-03);
   n=length (I);    n1=find (diff (I) >1);
   m1=fix (mean (I (1: n1)));    m2=fix (mean (I (n1+1: n)));
   text (4.5, 4, str2mat ( ['Y1 = ' num2str (round (Y (m1) *10000) *.1) …
        ',']、['Y2 = ' num2str (round (Y (m2) *10000) *.1) ',']), …
        'fontname','arial','fontsize', 9,'fonta','italic','color','r');
   L1=acot(imag(Y(m1)*Zc))/2/pi;   L1=round(L1*1000)*.001; L2=.5-L1;
   if L1<0; L1=L1+.5; end; if L2<0; L2=L2+.5; end;
```

```
    text (7, 4, str2mat ( ['L1 = ' num2str (L1)], ['L2 = ' num2str (L2)]), ...
        'fontname', 'arial', 'fontsize', 9, 'fonta', 'italic', 'color', 'r');
    text (8.4, 4, str2mat ('l', 'l'), 'fontname', 'symbol', 'fontsize', ...
        10, 'color', 'r');
    if m1<imax; m1=m1+500; m2=m2+500; end;
    text (2.5, 4, str2mat ( ['d1 = '...
        num2str (round (z (m1-imax+1) *1000) *.001)], ...
         ['d2 = ' num2str (round (z (m2-imax+1) *1000) *.001)]), ...
        'fontname', 'arial', 'fontsize', 9, 'fonta', 'italic', 'color', 'r');
    text (3.9, 4, str2mat ('l,', 'l,'), 'fontname', 'symbol', ...
        'fontsize', 10, 'color', 'r');
    zz1=10* (1-z (m1)); zz2=10* (1-z (m2)); zmax=10* (1-z (imax));
    plot ( [zz1 zz1], [-.1 1], 'r:', [zz2 zz2], [-.1 1], 'r:', ...
         [0 0] +zmax, [-.1 1], 'b:');
    text (zz1, .5, 'd1', 'fontname', 'arial', 'fontsize', 9, ...
        'fonta', 'italic', 'color', 'r');
text (zz2, .5, 'd2', 'fontname', 'arial', 'fontsize', 9, ...
        'fonta', 'italic', 'color', 'r');
end;
hold off;
```

图 5.39 为 trans.m 函数的运行结果。

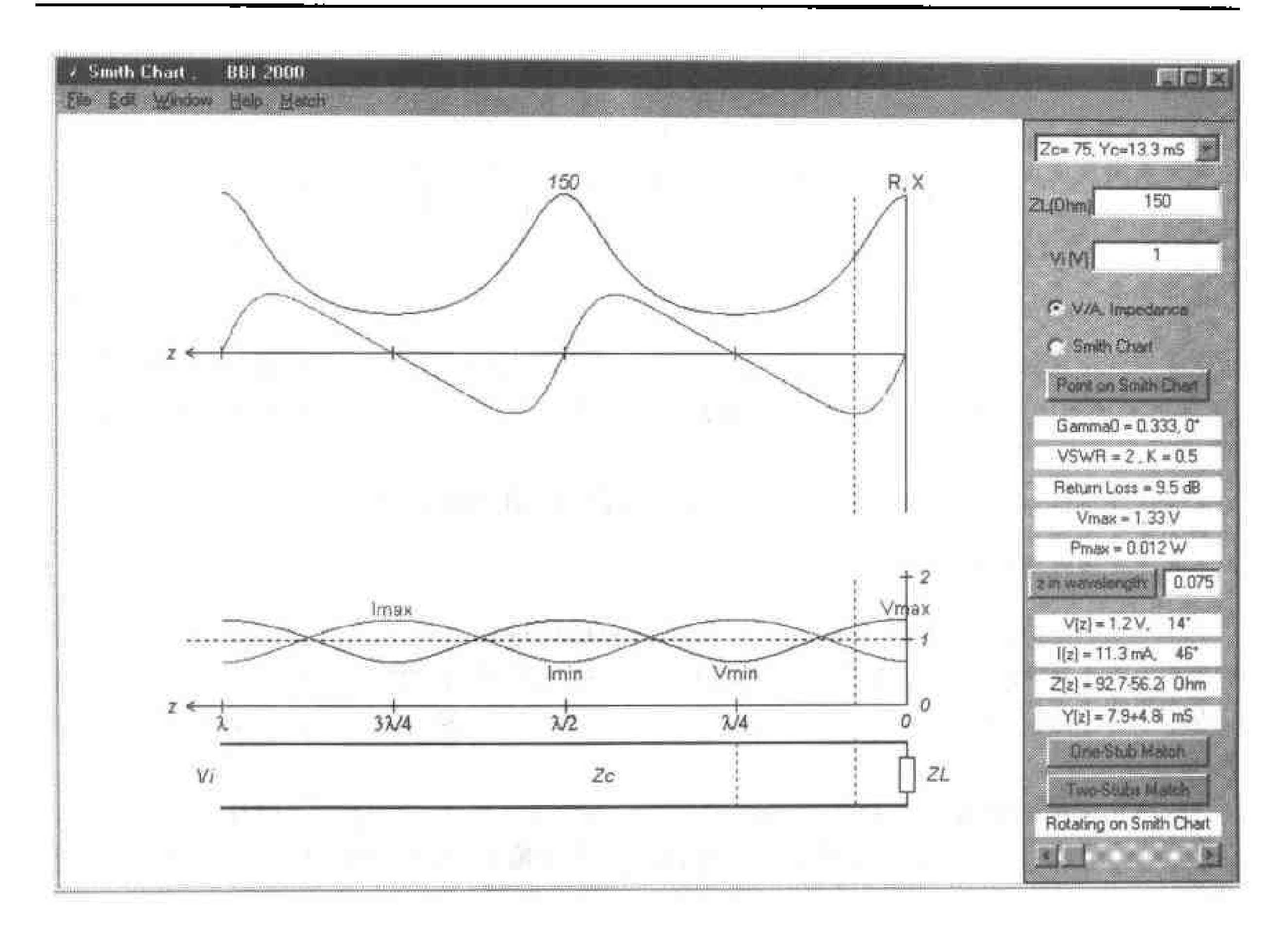


图 5.39 trans.m 函数的运行结果

参考文献

[1] 吴大正．信号与线性网络分析（上册）．北京：人民教育出版社，1980

[2] 李瀚荪．电路分析基础（下册）．北京：人民教育出版社，1979

[3] 龚汉民，丁宗豪．卫星电视接收与转播．北京：科学技术文献出版社，1991

[4] 秦绮玲，车晴．共用天线电视系统．北京：科学技术文献出版社，1991

[5] 蒋焕文等．低频电子线路．北京：人民铁道出版社，1979

[6] 贺允东等．高频电子线路．北京：人民铁道出版社，1979

[7] 应嘉年，顾茂章，张克潜．微波与光导波技术．北京：国防工业出版社，1994

第六章　模拟电子系统

在本章内通过列举实例的方式来介绍如何使用MATLAB对模拟电子系统进行仿真的问题。这些实例均是由本书作者自己编写的，并已经在北京广播学院的教学工作中使用过多年。

本章列举的各种实例中多数是使用4.2版MATLAB来进行的。

6.1　音频频谱分析仪

自定义M函数audspec.m用来对音频频谱分析仪进行仿真，它通过鼠标来选择一个声音文件（wav格式），它可播放该声音文件、显示输入音频信号的波形和频谱，同时还支持用鼠标来任意选择一条谱线，并给出该谱线所对应的音高，故audspec.m函数是一个比较实用的仿真软件。

WAV文件的输入方式采用了uigetfile函数，该函数打开一个标准的对话框，然后可以选择声音文件的路径和文件的名称，显然这样的文件输入方式比键盘输入的方式要方便得多。

信号波形生成使用绘图函数就可以完成，而信号的频谱是通过快速傅立叶变换而得到的。

谱线的音高是根据平均律来确定的，即半音的频率比为$2^{1/12}\approx1.0594631$，全音的频率比为$2^{1/6}\approx1.1224625$，八度音的频率比为2，另外我们知道a1（即简谱中的6）的频率是440Hz，于是以它基准就不难求出其它各音的频率。中央C到高音C之间各音的频率数值如图6.1所示。

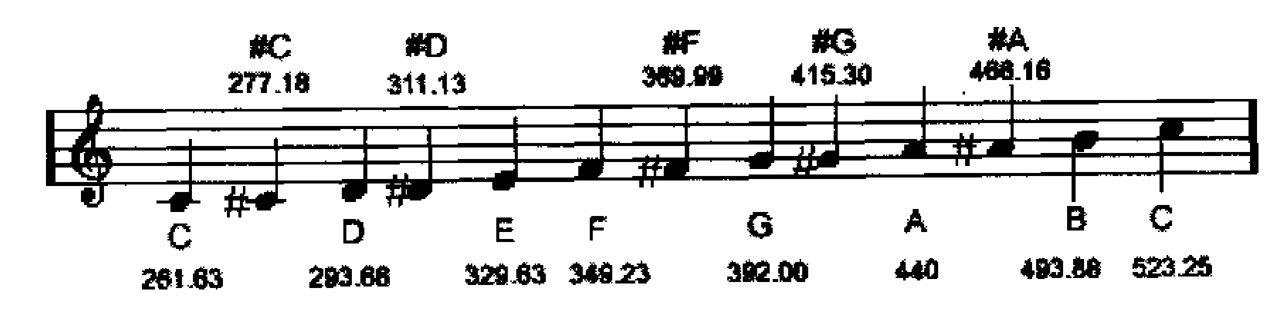


图6.1　音高与频率

audspec.m函数的使用方法为:》audspec;

键入audspec后，屏幕上出现图6.2所示的对话框，使用鼠标来选择要进行频谱分析的声音文件；程序先显示声音信号的波形，在该窗口的菜单栏内，可以选择play（播放）、spectrum（频谱）、zoom（放大）等功能；在频谱窗口的菜单栏内，可以继续选择find－scale

(谱线音高)、liner/log(线性/对数坐标)等功能。使用鼠标选择一条谱线(见图6.3中的十字线),然后便可确定其音高,结果显示在频谱窗口的标题栏内,如图6.3所示。使用鼠标可以在示波器窗口和频谱分析仪窗口之间进行切换。

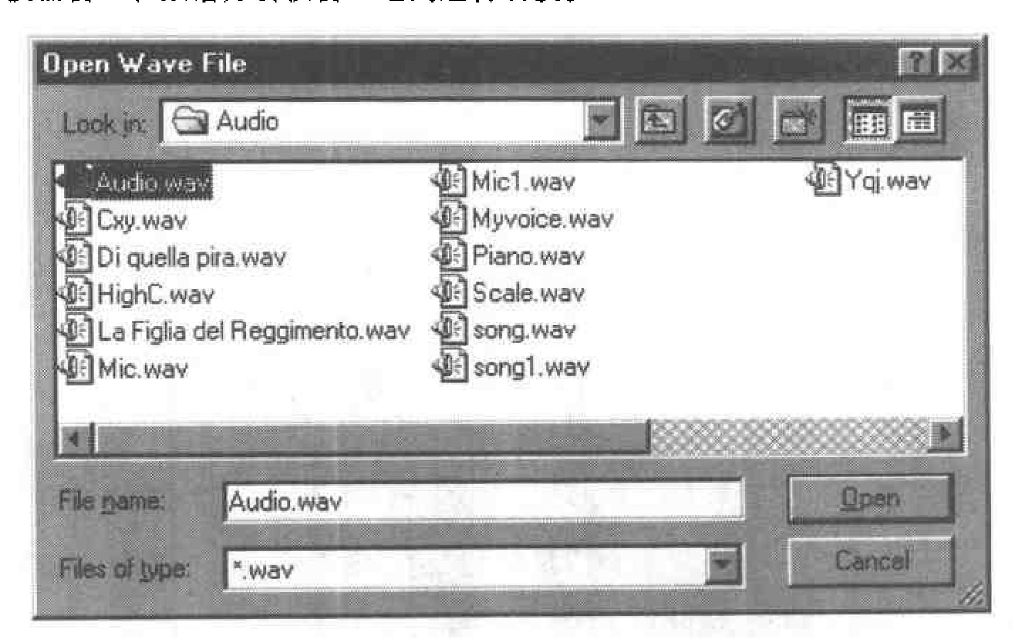


图6.2 文件选择对话框

audspec.m函数的程序清单如下:

```
function y=audspec (action);
%   AUDIO SPECTRUM ANALYSER                    BBI 99/12
%
%   Usage: y=audspec;
global x t dt fs c1 c2 H-osc H2 fn y Ktype
if nargin<1; action='initialized'; end;
if strcmp (action,'initialized');                 % 示波器
    clc; v=version;
     [fname, pname] =uigetfile ('*.wav','Open Wave File');
    file= [pname fname];
    if strcmp (v (1),'5');
         [x, fs, nbits] =wavread (file); [m, n] =size (x);
        file=[file ' fs=' int2str (fs) 'Hz, ' int2str (nbits) ...
            ' bits'];
        if n= =2; file= [file ', stero']; x=x (:); end;
    elseif strcmp (v (1),'4');
         [x, fs] =wavread (file);
        file= [file ' fs=' int2str (fs) 'Hz'];
```

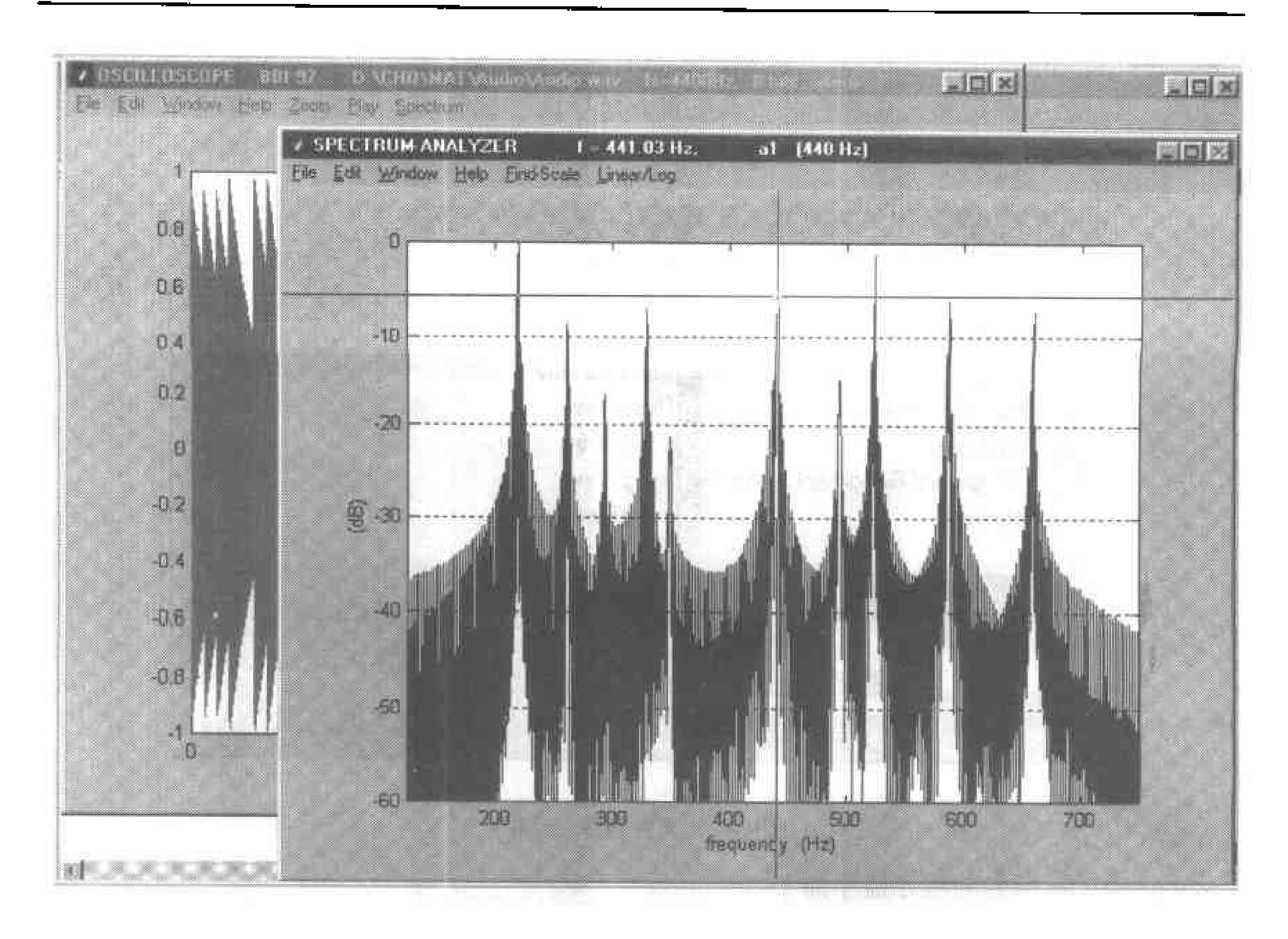


图 6.3 音频频谱分析仪（MATLAB 5.2 中的运行结果）

```
    end;

    H_osc = figure ('Name', ['OSCILLOSCOPE BBI 97' blanks (6) file], ...
        'NumberTitle','off','Position', [10 78 640 480]);

     uimenu ('Label','&Zoom','Callback','audspec (''Zoom'');');
     uimenu ('Label','&Play','Callback','audspec (''Play'');');
     uimenu ('Label','&Spectrum','Callback','audspec (''Spectrum'');');
if strcmp (v (1),'4');
     whitebg ('k'); c1 = 'y'; c2 = 'g';
elseif strcmp (v (1),'5');
     c1 = 'r'; c2 = 'b';
end;
dt = 1/fs; N = length (x); t = (0: N-1) * dt; T = N * dt; ss = 's';
if T<1e-03; t = t * 1e+06; ss = 'μs';
elseif T<1; t = t * 1000; ss = 'ms';
end;
```

```
plot (t, x, c1);
set (gca,'FontName','Arial','FontSize', 10);
   xlabel ( |'t (' ss ')']);
   n1 =1; n2 =N; H2= -1;
elseif strcmp (action,'Play');                    %  播放声音文件
     v=axis;                  del=t (2) -t (1);     del=del * .5;
     n1=find (abs (t-v (1)) <del);          n2=find (abs (t-v (2)) <del);
     if isempty (n2); n2=length (x); end;
     xx=x (n1: n2); sound (xx, fs);

elseif strcmp (action,'Zoom');                    %  放大
     zoom (H_osc,'xon');

elseif strcmp (action,'Spectrum');                %  频谱分析仪
     v=axis; del=t (2) -t (1); del=del * .5;
     n1=find (abs (t-v (1)) <del); n2=find (abs (t-v (2)) <del);
     if isempty (n2); n2=lengtb (x); end;

     if H2>0; delete (H2); end;
   H2=figure ('Name','SPECTRUM ANALYZER BBI 97', ...
       'NumberTitle','off','Position', [155 32 640 480]);

    uimenu ('Label','&Find-Scale','Callback','audspec (''Scale'');');
    uimenu ('Label','&Linear/Log','Callback','audspec (''Log'');');

  xx=x (n1: n2);            N=length (xx);
     f1=1/ (dt*N);           % frequency discrimination
     y=fft (xx/N);           y=abs (y) *2;
     N2=fix (N/2);           y=y (1: N2); y=y/max (y);
     I=find (y<1e-3);        y (I) =1e-3*ones (size (I));
  y=20*log10 (y);

     fn=f1* (0: N2-1);   I=find (fn>20);      fn=fn (I); y=y (I);
     plot (fn, y, c2);
   ystr=' (dB)';          Ktype= -1;
   set (gca,'Ygrid','on','YLim', [-60 0], ...
       'FontName','Arial','FontSize', 10);
     xlabel ( ['frequency (' 'Hz)']); ylabel (ystr); zoom xon;
```

```
  elseif strcmp (action,'Log'); % 线性/对数坐标 (频率)
     if Ktype = = -1;
        semilogx (fn, y, c2);
     elseif Ktype = =1;
        plot (fn, y, c2);
     end;           Ktype = -Ktype;
       set (gca,'Ygrid','on','YLim', [-60 0], ...
       'FontName','Arial','FontSize', 10);
       xlabel ( ['frequency (' 'Hz)']); ylabel (' (dB)'); zoom xon;

elseif strcmp (action,'Scale'); % 确定音高
     [u, v] =ginput (1); u=round (u*100) *.01; note=frq2not (u);
     f=not2frq (note); f=round (f*100) *.01;
     ntstr= [note ' (' num2str (f) ' Hz)'];
     set (H2,'Name', ...
       ['SPECTRUM ANALYZER' blanks (10) 'f = ' num2str (u) ...
       ' Hz, ' ntstr]);

end;
```

需要说明的是，在进行频谱分析的过程中数据不能太多，否则频谱分析的时间会过长。数据量一般控制在一万以内，这可以通过示波器窗口的 zoom 菜单选择一个比较合适的时间区间来进行频谱分析。

6.2 幅度调制

幅度调制为一种广泛使用的模拟调制方式，简称调幅（AM），由于其载波的幅度随调制信号作线性变化，因此幅度调制又称为线性调制。幅度调制又可以进一步分为调幅（AM）、双边带（DSB）、单边带（SSB）、残留边带（VSB）和平衡正交调幅（QM）等几种方式。

设调制信号为 m (t)，其幅度为 1V，载波信号为 sinωt，角频率为 ω，各种调幅方式的已调波的表达式如下：

调幅：　$s(t) = [1+m_A \cdot m(t)] \cdot \sin\omega t$，其中调制度 m_A 的数值范围在 0～1 之间。

双边带：$s(t) = m(t) \cdot \sin\omega t$

单边带：$s(t) = 0.5 \cdot m(t) \cdot \cos\omega t + 0.5 \cdot \hat{m}(t) \cdot \sin\omega t$

$s(t) = 0.5 \cdot m(t) \cdot \cos\omega t - 0.5 \cdot \hat{m}(t) \cdot \sin\omega t$

$\hat{m}(t)$ 为调制信号 m (t) 的希尔伯特变换。

平衡正交调幅：$s(t) = m_1(t) \cdot \sin\omega t + m_2(t) \cdot \cos\omega t$

幅度调制解调器可以分为包络检波和同步解调两种基本方式，包络检波适用于普通调幅和残留边带调幅，而同步解调适用于各种幅度调制方式。包络检波器的结构十分简单，它包

括一个（或两个）检波二极管和一个低通滤波器（由一个电阻和一个电容组成）。同步解调器的核心部分是由一个乘法器和一个低通滤波器构成，由于同步解调要求在接收端必须产生一个与输入载波信号同步的本地载波信号，因此同步解调器中还必须设有一个载波恢复装置，而该装置的结构是十分复杂的。

6.2.1 调幅波的模块仿真

在本书中，模块仿真是特指使用 simulink 工具箱和 DSP 工具箱进行的仿真。模块仿真给出了系统的方框图，概念特别清楚，同时与实际系统比较接近，可以观察系统中个点的波形和频谱，另外采用模块仿真是不需要进行编程的。

普通调幅和双边带调幅的模块仿真如图 6.4 所示，在仿真过程中使用了 SIMULINK 工具箱和 DSP 工具箱，具体步骤如下：

首先建立系统。在 MATLAB 环境下键入 simulink 和 dsplib，分别打开系统仿真的模块库和 DSP 模块库。在 SIMULINK 的源模块库中找到信号发生器模块，在 SIMULINK 的输出模块库内找到示波器模块，在 SIMULINK 的线性模块库内找到加法器模块，在 SIMULINK 的非线性模块库内找到乘法器模块，在 SIMULINK 的接口模块库内找到复接器模块，在 DSP 的输出模块库内找到带缓存的 FFT 示波器模块（频谱仪），然后参照图 6.4 将各个模块连接起来。使用鼠标的右键点击模块，将信号发生器、示波器和频谱仪设置为青色（cyan），而将其它模块设置成为黄色，以示区别。

系统建立之后，还需对各个环节的具体参数进行设置。调制信号发生器的参数设置：频率为 1Hz，信号幅度为 1V，信号类型为正弦波。载波信号发生器的参数设置：频率为 10Hz，信号幅度为 1V，信号类型为正弦波。AM－DSB 示波器的参数设置：y 轴最大值为 1.2V，y 轴最小值为－1.2V，显示时间范围为 2s，采样时间间隔为 0.0025s（采样频率为 400Hz）。AM－DSB－TC 示波器的参数设置：y 轴最大值为 2.2V，y 轴最小值为－2.2V，显示时间范围为 2s，采样时间间隔为 0.0025s（采样频率为 400Hz）。频谱仪的参数设置：采样时间间隔为 0.025s（采样频率为 40Hz），其它参数均使用缺省值。在模型窗口的 Simulation 菜单中找到 Parameters 子菜单，设定系统仿真的模式为：VariableStepDiscrete。最后，使用鼠标点击 File 菜单内的 Save 子菜单，将整个系统储存在一个名为 AMDSB_TC.dml 的文件内。

在 MATLAB 的工作环境下，键入 amdsb_tc 就可以进行调幅波生成环节的系统仿真了，其结果见图 6.4。从示波器可以看出，普通调幅波（品红色的曲线）的包络与调制信号（黄色的曲线）是完全一样的，而双边带调幅的包络则为调制信号的绝对值。频谱仪清楚地显示出单频信号调制时，普通调幅波的频谱共有三条，一条是载波，另外两条为上下边带，边带的幅度比载波的幅度低 6dB，边带与载波的频率之差恰好为调制信号的频率。

在调制信号发生器中还可以选择方波信号或锯齿波信号作为调制信号，然后点击 Start 按钮便可对这两种调制信号进行系统仿真。

6.2.2 幅度调制的仿真函数

使用模块仿真可以清楚地显示出系统整体情况，而若想对调幅的各项参数进行细致的分析则应该使用仿真函数。

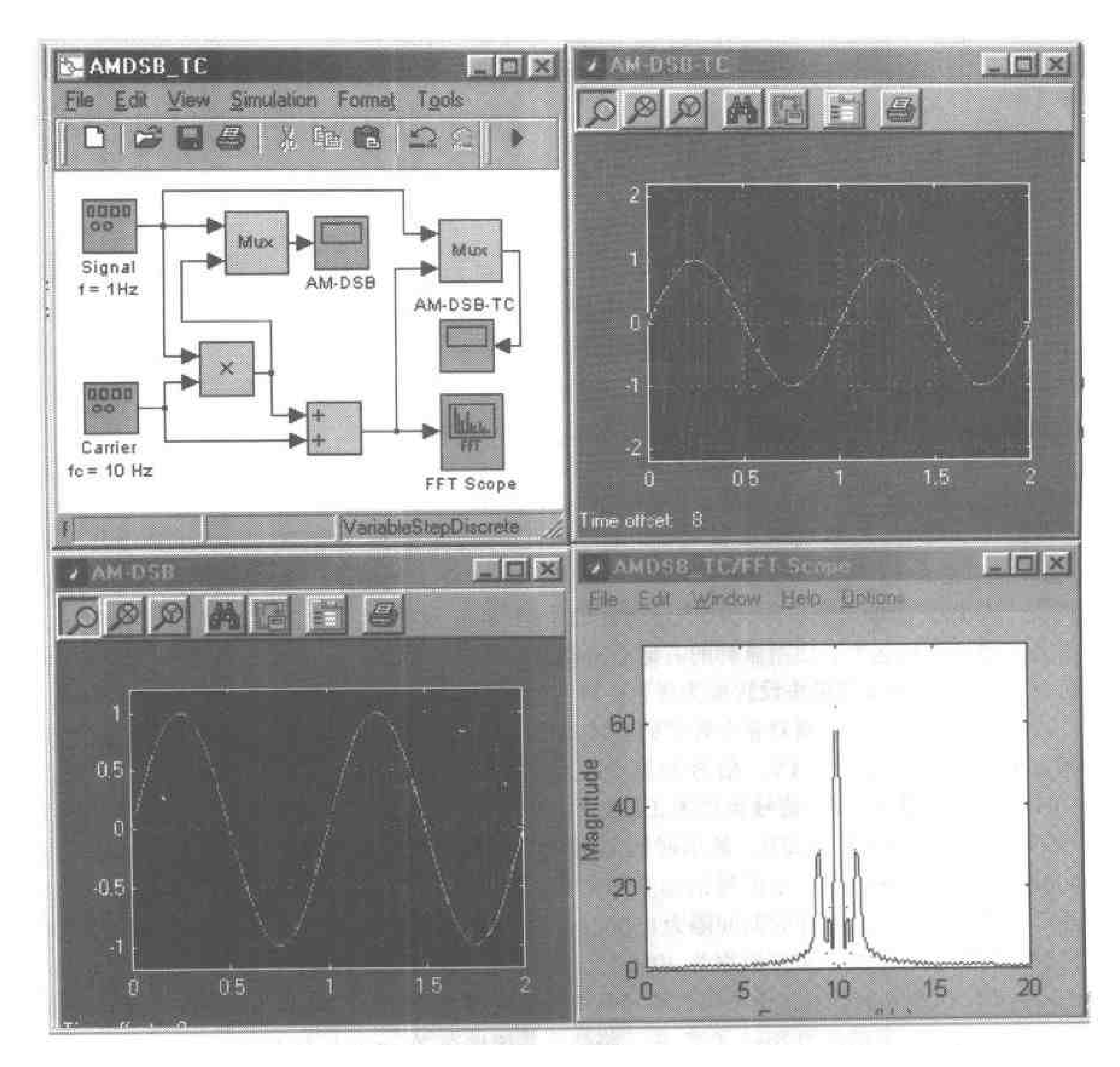


图 6.4 调幅波的模块仿真

自定义的 M 函数 am.m 用来对幅度调制进行仿真，它在 4.2 版和 5.x 版内均可运行。为了便于使用，在函数中采用了用户控制框来输入各种参数。调制方式有：AM、DSB、SSB 下边带、SSB 上边带。调制信号类型有：正弦波、方波（脉冲）、锯齿波、狄里赫莱函数。普通调幅的调制度有：100%、90%、87.5%、80%、70%、60%、50%。调制信号和载波信号的频率任选。在调制过程中，屏幕显示调制信号的波形、载波信号的波形和频谱；在解调过程中，屏幕上分别显示调制信号、载波信号和解调信号的波 形。

在信号调制的过程中，根据各种调幅信号的数学模型来生成载波信号，在解调过程中，则使用了信号工具箱内的 demod 函数。

am.m 函数的运行过程中调用 am1.m 函数，具体的程序清单如下：

```
function [] =am (action);
```

```
%
% BBl 2000
global y x t fc fs f tstr Kw Kmod m H1 H2 H3 H42 H52
v=version; if nargin<1; action='initialized'; end;

if strcmp (action,'initialized');
    if strcmp (v (1),'5');                                % 5.x 版
        figure ('Num','off','Units','pix','pos', ...
            [5 29 792 530],'Color', [1 1 1]);
    elseif strcmp (v (1),'4');                            % 4.2 版
        figure ('num','off','units','pix','pos', [5 29 792 530]);
        whitebg ('w');
    end;

    uicontrol ('Style','Frame','Units','Normal','Position', ...
            [.82 0 .2 1],'Back', [.8 .8 .8]);

    H1=uicontrol ('Style','Popup','Units','Normal','Position', ...
            [.84 .9 .15 .05],'String', str2mat ('AM','AM-DSB', ...
            'AM-SSB Lower Band','AM-SSB Upper Band'), ...
            'Back', [1 1 1],'Callback','am method;');

    H2=uicontrol ('Style','Popup','Units','Normal','Position', ...
            [.84 .8 .15 .05],'String', str2mat ('Sine', ...
            'Square','Sawtooth','Diric'), ...
            'Back', [1 1 1],'Callback','am wave;');

    H3=uicontrol ('Style','Popup','Units','Normal','Position', ...
            [.84 .7 .15 .05],'String', str2mat ('100%', ...
            ' 90%','87.5%',' 70%',' 60%',' 50%'), ...
            'Back', [1 1 1],'Callback','am mr;');

    H41=uicontrol ('Style','Text','Units','Normal','Position', ...
       [.83 .595 .05 .04],'String','f (Hz):','Back', [.8 .8 .8]);

    H42=uicontrol ('Style','Edit','Units','Normal','Position', ...
            [.89 .6 .1 .04],'Back', [1 1 1],'String', ...
            '15.625e+03','Call','am f;');
```

```
    H51 = uicontrol ('Style', 'Text', 'Units', 'Normal', 'Position', ...
      [.822 .495 .06 .04], 'String', 'fc (Hz):', 'Back', [.8 .8 .8]);

    H52 = uicontrol ('Style', 'Edit', 'Units', 'Normal', 'Position', ...
      [.89 .5 .1 .04], 'Back', [1 1 1], 'String', '38e+06', 'Call', 'am fc;');

    H6 = uicontrol ('Style', 'Push', 'Units', 'Normal', 'Position', ...
      [.84 .4 .15 .05], 'String', 'RESTORE', 'Call', 'am restore;');
    H7 = uicontrol ('Style', 'Push', 'Units', 'Normal', 'Position', ...
      [.84 .3 .15 .05], 'String', 'DEMOD', 'Call', 'am demod;');
    f = 15.625e+03; fc = 38e+06; m = 1; Kw = 1; Kmod = 1;
    [y, x, t, fc, fs, tstr] = am1 (f, fc, m, Kmod, Kw, 1);

elseif strcmp (action, 'method');                         % 调制方式
    Kmod = get (gco, 'Value');
    [y, x, t, fc, fs, tstr] = am1 (f, fc, m, Kmod, Kw, 1);
    if Kmod>1; set (H3, 'value', 1); end;

elseif strcmp (action, 'wave');                           % 调制信号类型
    Kw = get (gco, 'Value');
    [y, x, t, fc, fs, tstr] = am1 (f, fc, m, Kmod, Kw, 1);

elseif strcmp (action, 'mr');                             % 调制度
    k = get (gco, 'Value'); mt = [1 .9 .875 .7 .6 .5]; m = mt (k);
    if Kmod = = 1;
        [y, x, t, fc, fs, tstr] = am1 (f, fc, m, Kmod, Kw, 1);
    else;
        set (H3, 'value', 1);
    end;

elseif strcmp (action, 'f');                              % 调制信号频率
    f = get (gco, 'string'); f = str2num (f);
    [y, x, t, fc, fs] = am1 (f, fc, m, Kmod, Kw, 1);
elseif strcmp (action, 'fc');                             % 载波信号频率
    fc = get (gco, 'string'); fc = str2num (fc);
    [y, x, t, fc, fs, tstr] = am1 (f, fc, m, Kmod, Kw, 1);
elseif strcmp (action, 'restore');                        % 复原
    f = 15.625e+03; fc = 38e+06; m = 1; Kw = 1; Kmod = 1;
    [y, x, t, fc, fs, tstr] = am1 (f, fc, m, Kmod, Kw, 1);
```

```
    set (H42,'string','15.625e+03'); set (H52,'string','38e+06');
    set (H1,'value', 1); set (H2,'value', 1); set (H3,'value', 1);

elseif strcmp (action,'demod');                                  % 解调
    x1=demod (y, fc, fs,'am'); x1=x1-mean (x1);
    if strcmp (v (1),'4');
        subplot (1, 1, 1); set (gca,'vis','off');
        subplot ('position', [.065 .5811 .7 .3439]);
        plot (t, y,'b', t, x,'r'); zoom xon; xlabel ( ['t (' tstr ')']);
    end;
    subplot ('position', [.065 .11 .7 .3439]);
    i=3: length (t) -8; plot (t (i), x1 (i),'b');
    xlabel ( ['t (' tstr ')']); title ('Demodulated Signal');

end;
```

```
function [y, x, t, fc, fs, tstr] =am1 (f, fc, m, Kmod, Kw, Ks);
%
% BBI 2000
if nargin<6; Ks=0;            end;
if nargin<5; Kw=1;            end;
if nargin<4; Kmod=1;          end;
if nargin<3; m=1;             end;
if nargin<2; fc=38e+06;       end;
if nargin<1; f=15.625e+03;    end;
fs=4*fc; dt=1/fs; T=2/f; t=0: dt: T-dt; f1=1/T;
if Kw==1;
    x=sin (2*pi*f*t-pi/2);
elseif Kw==2;
    x=square (2*pi*f*t, 10); m=m- (m==1) *eps;
elseif Kw==3;
    x=sawtooth (2*pi*f*t, .98);
elseif Kw==4;
    x=diric (2*pi*f*t, 32);
end;
if Kmod==1;
    y= (1+m*x) .*cos (2*pi*fc*t); astr='AM,';
```

```
elseif Kmod= =2;
     y=x. * cos (2 * pi * fc * t); astr='AM-DSB,';
elseif Kmod= =3;
     y=x. * cos (2 * pi * fc * t) +imag ( (hilbert (x))) . * sin (2 * pi * fc * t);
     astr='AM-SSB lower band,';
     I=find (y>1.1); y (I) =ones (size (I)); I=find (y<-1.1);
     y (I) =-ones (size (I));
elseif Kmod= =4;
     y=x. * cos (2 * pi * fc * t) -imag ( (hilbert (x))) . * sin (2 * pi * fc * t);
     astr='AM-SSB upper band,';
     I=find (y>1.1); y (I) =ones (size (I)); I=find (y<-1.1);
     y (I) =-ones (size (I));
end;
if fc>=1e+06;
     t=t * 1e+06; f1=f1/1e+06; tstr='μs'; fstr='MHz'; Kf=1e-06;
elseif fc>=1000;
     t=t * 1000; f1=f1/1000; tstr='ms'; fstr='kHz'; Kf=1e-03;
else;
     tstr='s'; fstr='Hz'; Kf=1;
end;
v=version;
if strcmp (v (1),'4');
     subplot (1, 1, 1); set (gca,'vis','off');
end;
if Ks= =1; subplot ('position', [.065 .5811 .7 .3439]);
else; subplot (211); end;
plot (t, y,'b', t, x,'r'); zoom xon; xlabel ( ['t (' tstr ')']);
n=length (y); yy=abs (fft (y) /n); yy=yy (1: fix (n/2)); ymax=max (yy);
yy=yy/ymax; yy=20 * log10 (yy);
I=find (yy>-40); fn= (I-1) * f1; yn=yy (I); m=length (I);
if Ks= =1; subplot ('position', [.065 .11 .7 .3439]);
else; subplot (212); end;
for i=1: m;
     plot ( [0 0] +fn (i), [-40 yn (i)],'b'); hold on;
end;
xlabel ( ['f (' fstr ')']); ylabel ('dB'); v=axis; zoom xon;
dv= (v (2) -v (1)) * .05; axis ( [v (1) -dv v (2) +dv v (3: 4)]); hold off;
set (gcf,'num','off','Name', [astr blanks (10) 'BBI 2000']);
```

am 函数的运行结果如图 6.5 所示。缺省的调制信号频率为 15625Hz，载波信号频率为 38MHz。在双边带、单边带方式下，调制度均为 100%。正弦波调制的频谱为三条谱线，锯齿波调制的频谱为若干条幅度递减的谱线，方波（脉冲）的频谱为若干簇密集分布的谱线，而狄里赫莱函数调制的频谱为若干条幅度相等的谱线。在调制信号发生突变的地方，如锯齿波的起始处、脉冲的前沿和后沿处，解调信号会产生一定的过冲失真，这与实际情况是吻合的。

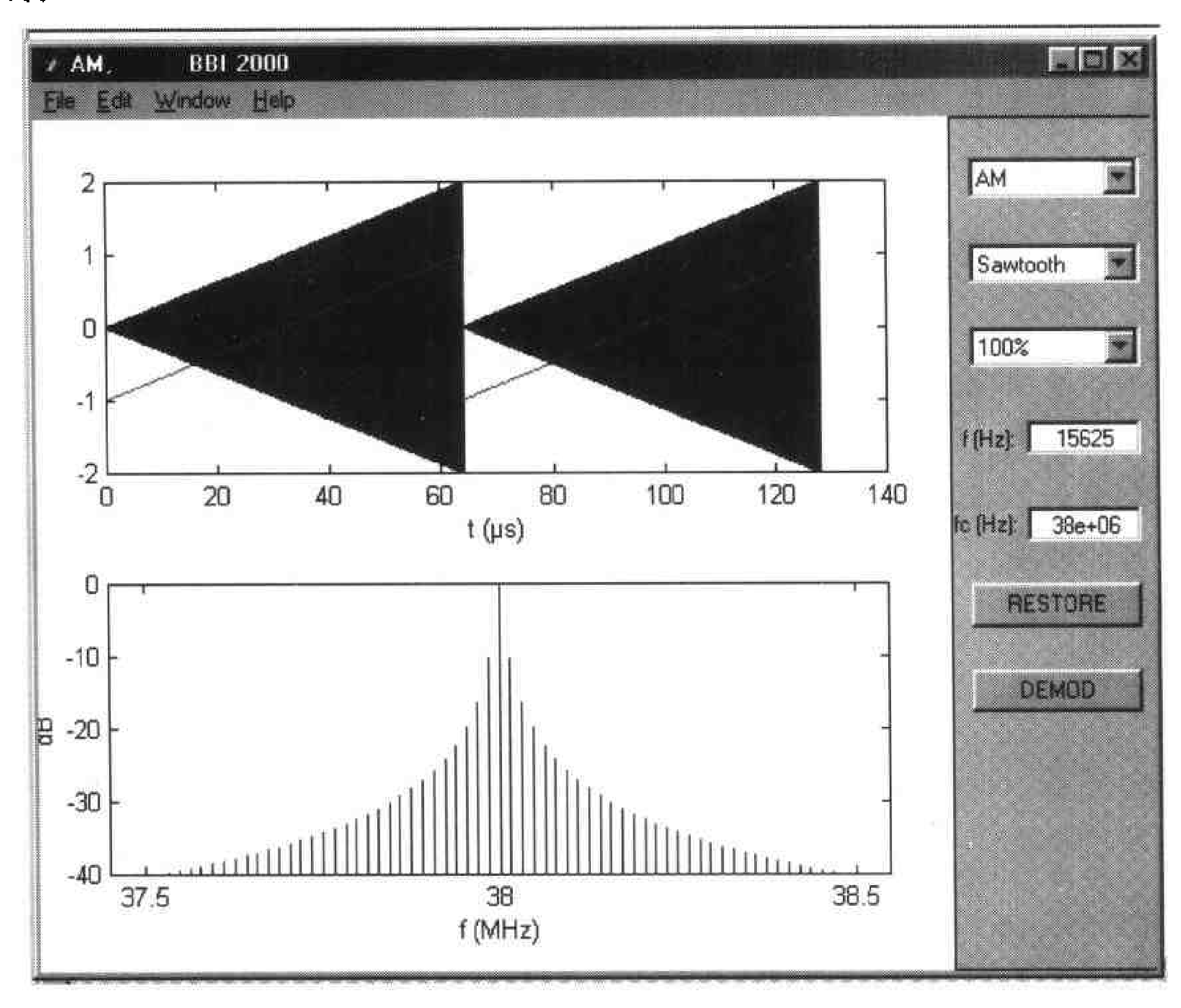


图 6.5 幅度调制与解调

6.2.3 平衡正交调幅与解调的模块仿真

平衡正交调幅的特点是可以在同一载波频率上调制两路不同的信号，其中一路信号调制在正弦波上，而另一路信号则调制在余弦波上；平衡正交调幅的解调采用同步解调的方式。在模拟电视广播中，两路色差信号的传输就是采用平衡正交调幅的方式。

平衡正交调幅与解调的仿真采用模块仿真的方式，整个系统的概念是十分清晰的。图 6.6 为模块仿真的结构与结果，从中可以观察到两路调制信号的波形，载波信号的波形和频

谱，两路解调信号的波形。

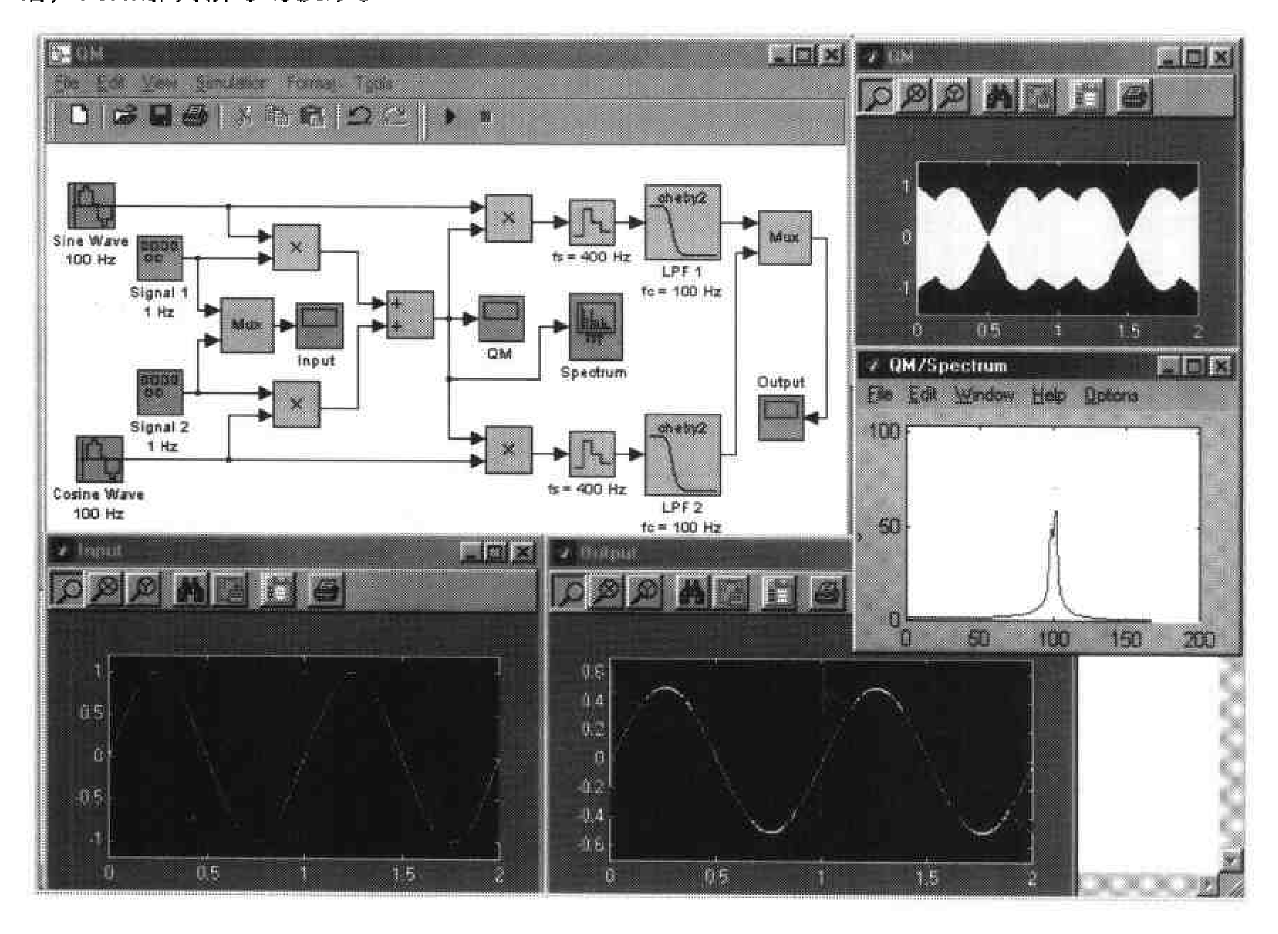


图 6.6　平衡正交调幅与解调

系统的建立过程与前述的调幅波的模块仿真系统建立的过程是类似，只是增加了两个正弦波信号发生器、两个数字低通滤波器和与之配套的采样模块，正弦波信号发生器位于 SIMULINK 工具箱的源模块库内，数字滤波器模块位于 DSP 工具箱的滤波器设计模块库内，而采样模块则位于 SIMULINK 工具箱的离散模块库内。系统储存在 qm.mdl 文件中。

信号发生器的参数设置：频率为 1Hz，信号幅度为 1V，调制信号的类型分别为正弦波和锯齿波。正弦载波发生器的设置：角频率为 200π（频率为 100Hz），相位为 0，信号幅度为 1V，采样时间间隔为 0.0025s（采样频率为 400Hz）。余弦载波发生器的设置：角频率为 200π（频率为 100Hz），相位为 π/2，信号幅度为 1V，采样时间间隔为 0.0025s（采样频率为 400Hz）。各个示波器的显示时间范围均为 2s，Input 示波器的 y 轴设置为 1.2V 和 −1.2V，QM 示波器的 y 轴设置为 1.5V 和 −1.5V，Output 示波器的 y 轴设置为 0.6V 和 −0.6V。频谱仪的采样时间间隔设定为 0.0025s，其余的均采用缺省参数。采样模块的采样时间间隔设定为 0.0025s。滤波器采用 4 阶的切比雪夫 II 型低通滤波器，通带的转折频率归一值为 0.5（频率为 100Hz），阻带衰减为 40dB。系统仿真的方式设定为 VariableStepDiscrete。

在 MATALB 工作环境下，键入 qm 便可对平衡正交调幅和解调的过程进行仿真，具体的仿真结果见图 6.6，显然解调信号与调制信号是十分接近的，而载波信号的包络为调制信

号包络（两路调制信号中幅度较大者）的绝对值。另外，还可选择方波信号或随机信号进行仿真。

6.3 角度调制

角度调制又称为非线性调制，其特点是：载波信号的幅度恒定，载波信号的频率或相位随调制信号变化，也就是说已调波的频谱与调制信号的频谱之间存在着非线性变换关系。角度调制又可进一步分为频率调制和相位调制两种方式，频率调制简称为调频（FM），相位调制简称为（PM）。

角度调制的一般表达式为：

$$s(t) = \sin[\omega t + \theta(t)]$$

其中 θ（t）称为瞬时相位。

对于相位调制来说，瞬时相位正比与调制信号 m（t），即

$$\theta(t) = K_P \cdot m(t)$$

其中 K_P 为一个常数。相位调制的峰值相位偏移（最大相移）称为相位调制指数，其定义为：

$$m_P = K_P \cdot \max[m(t)]$$

对于频率调制来说，瞬时相位正比于调制信号的积分，即

$$\theta(t) = K_F\int_{\infty}^{t} m(\tau)d\tau$$

其中 K_F 为一个常数。瞬时角频率的频偏可以表示为相位的微分，即 $\omega - \omega_0 = K_F \cdot m\ (t)$，而瞬时的频偏则可以表示为：$f - f_0 = \frac{K_F}{2\pi} \cdot m\ (t)$，于是峰值频率偏移（最大频偏）可以表示为 $\Delta F = \frac{K_F}{2\pi}\max\ [m\ (t)]$。频率调制的调制指数定义为：

$$m_F = \frac{\Delta F}{B} = \frac{K_F}{2\pi B} \cdot \max[m(t)]$$

其中 B 为调制信号的带宽，单位为 Hz。

自定义的 M 函数 fmpm.m 用来对角度调制进行仿真，它在 4.2 版和 5.x 版内均可以运行。为了便于使用，在函数中采用了用户控制框来输入各种参数。调制方式有：AM 和 PM，在频率调制时，需要对频偏进行设置，而在相位调制时，则需要对调制指数进行设置。调制信号类型有：正弦波、方波（脉冲）、锯齿波、狄里赫莱函数。调制信号和载波信号的频率任选。在调制过程中，屏幕显示调制信号的波形、载波信号的波形和频谱；在解调过程中，屏幕上分别显示调制信号、载波信号和解调信号的波形。

在信号调制的过程中，根据角度调制的数学模型来生成载波信号，并使用了累加的方式替代了调频模型中出现的积分，而在解调过程中，则使用了信号工具箱内的 demod 函数，并进一步对解调信号进行了数字滤波。

fmpm.m 函数的运行过程中调用 fmpm1.m 函数，具体的程序清单如下：

```
function [ ] = fmpm (action);
%
```

```
% BBI 2000;
global y x t fc fs f tstr Kw Kmth K H1 H21 H22 H3 H42 H52
v=version; if nargin<1; action='initialized'; end;

if strcmp (action,'initialized');
    if strcmp (v (1),'5');
        figure ('Num','off','Units','pix','pos', [5 29 792 530], ...
            'Color', [1 1 1]);
    elseif strcmp (v (1),'4');
        figure ('num','off','units','pix','pos', [5 29 792 530]);
        whitebg ('w');
    end;

    uicontrol ('Style','Frame','Units','Normal','Position', ...
            [.82 0 .2 1],'Back', [.8 .8 .8]);

    H1=uicontrol ('Style','Popup','Units','Normal','Position', ...
            [.84 .9 .15 .05],'String', str2mat ('FM','PM'),'Back', ...
            [1 1 1],'Callback','fmpm method;');

    H21=uicontrol ('Style','Text','Units','Normal','Position', ...
        [.822 .84 .18 .03],'String',' Frequency Deviation (Hz)', ...
        'Back', [.8 .8 .8]);

    H22=uicontrol ('Style','Edit','Units','Normal','Position', ...
        [.84 .8 .15 .04],'Back', [1 1 1],'String', ...
        '10e+03','Call','fmpm K;');

    H3=uicontrol ('Style','Popup','Units','Normal','Position', ...
        [.84 .7 .15 .05],'String', str2mat ('Sine','Square-I', ...
        'Square-II','Sawtooth-I','Sawtooth-II','Diric'),'Back', ...
        [1 1 1],'Callback','fmpm wave;');

    H41=uicontrol ('Style','Text','Units','Normal','Position', ...
        [.83 .595 .05 .04],'String','f (Hz):','Back', [.8 .8 .8]);

    H42=uicontrol ('Style','Edit','Units','Normal','Position', ...
      [.89 .6 .1 .04],'Back', [1 1 1],'String','2e+03','Call','fmpm f;');
```

```
    H51 = uicontrol ('Style','Text','Units','Normal','Position', ...
        [.822 .495 .06 .04],'String','fc (Hz):','Back', [.8 .8 .8]);

    H52 = uicontrol ('Style','Edit','Units','Normal','Position', ...
        [.89 .5 .1 .04],'Back', [1 1 1],'String','2e+04','Call','fmpm fc;');

    H6 = uicontrol ('Style','Push','Units','Normal','Position', ...
        [.84 .4 .15 .05],'String','RESTORE','Call','fmpm restore;');
    H7 = uicontrol ('Style','Push','Units','Normal','Position', ...
        [.84 .3 .15 .05],'String','DEMOD','Call','fmpm demod;');

    f=2e+03; fc=2e+04; Kw=1; Kmth=1; K=.5*fc;
    [y, x, t, fc, fs] =fmpm1 (f, fc, Kmth, Kw, K, 1);

elseif strcmp (action,'method');
    Kmth=get (gco,'Value');
    if Kmth==1;
        set (H21,'string',' Frequency Deviation (Hz)');
    elseif Kmth==2;
        set (H21,'string','PM Modulation Index');
        set (H22,'string','1'); K=1;
    end; [y, x, t, fc, fs, tstr] =fmpm1 (f, fc, Kmth, Kw, K, 1);

elseif strcmp (action,'wave');
    Kw=get(gco,'Value'); [y,x,t,fc,fs,tstr]=fmpm1(f,fc,Kmth,Kw,K,1);

elseif strcmp (action,'K');
    K=get (gco,'string'); K=str2num (K);
    [y, x, t, fc, fs] =fmpm1 (f, fc, Kmth, Kw, K, 1);

elseif strcmp (action,'f');
    f=get (gco,'string'); f=str2num (f);
    [y, x, t, fc, fs] =fmpm1 (f, fc, Kmth, Kw, K, 1);
elseif strcmp (action,'fc');
    fc=get (gco,'string'); fc=str2num (fc);
    [y, x, t, fc, fs, tstr] =fmpm1 (f, fc, Kmth, Kw, K, 1);
elseif strcmp (action,'restore');
    f=2e+03; fc=2e+04; Kmth=1; Kw=1; K=10e+03;
    set (H42,'string','2e+03'); set (H52,'string','2e+04');
```

```
    set (H21,'string',' Frequency Deviation (Hz)');
    set (H22,'string','10e+03'); set (H1,'value', 1);
    [y, x, t, fc, fs, tstr] =fmpml (f, fc, Kmth, Kw, K, 1);

elseif strcmp (action,'demod');
    mstr= ['fm';'pm'];
    x1=demod (y, fc, fs, mstr (Kmth,:)); x1=x1-mean (x1);
    [b, a] =chehy2 (4, 40, 1/30);
    x1=filter (b, a, x1);
    if strcmp (v (1),'4');
        subplot (1, 1, 1); set (gca,'vis','off');
        subplot ('position', [.065 .5811 .7 .3439]);
        plot (t, y,'b', t, x,'r'); zoom xon; xlabel ( ['t (' tstr ')']);
    end;
    subplot ('position', [.065 .11 .7 .3439]);
    i=1: length (t) -0; plot (t (i), x1 (i),'h');
    xlabel ( ['t (' tstr ')']); title ('Demodulated Signal');

end;
```

```
function [y, x, t, fc, fs, tstr] =fmpml (f, fc, Kmtb, Kw, K, Ks)
%
% y=sin (2*pi*fc*t+Kf*cumsum (x) /fs);          FM
% y=sin (2*pi*fc*t+Kp*x);                        PM
if nargin<4; Kw=1; end;
if nargin<3; Kmth=1; end;
if nargin<2; fc=2e+04; end;
if nargin<1; f=2e+03; end;
if nargin<5; K=.5*fc; end;
if nargin<6; Ks=0; end;
fs=50*fc; dt=1/fs; T=2/f; t=0: dt: T-dt; f1=1/T; N=length (t);
if Kw==1;
      x=sin (2*pi*f*t);
elseif Kw==2;
      x=square (2*pi*f*t, 10);
elseif Kw==3;
      x=square (2*pi*f*t);
```

```
elseif Kw==4;
    x=sawtooth (2*pi*f*t, .98);
elseif Kw==5;
    x=sawtooth (2*pi*f*t, .5);
elseif Kw==6;
    x=diric (2*pi*f*t, 32);
end;
if Kmth==1;
    dF=K; Kf=2*pi*dF/max (abs (x)); mf=dF/f;
    y=sin (2*pi*fc*t+Kf*cumsum (x) *dt); astr='FM,';
elseif Kmth==2;
    Kp=K; y=sin (2*pi*fc*t+Kp*x); mp=Kp*max (abs (x)); astr='PM,';
end;
if fc>=1e+06;
    t=t*1e+06; f1=f1/1e+06; tstr='μs'; fstr='MHz'; Kf=1e-06;
elseif fc>=1000;
    t=t*1000; f1=f1/1000; tstr='ms'; fstr='kHz'; Kf=1e-03;
else;
    tstr='s'; fstr='Hz'; Kf=1;
end;
v=version;
if strcmp (v (1),'4');
    subplot (1, 1, 1); set (gca,'vis','off');
end;
if Ks==1; subplot ('position', [.065 .5811 .7 .3439]);
else; subplot (211); end;
plot (t, y,'b', t, x,'r'); zoom xon; xlabel ( ['t (' tstr ')']);
yy=abs (fft (y) /N); yy=yy (1: fix (N/2)); ymax=max (yy);
yy=yy/ymax; yy=20*log10 (yy);
I=find (yy>-40); fn= (I-1) *f1; yn=yy (I); m=length (I);
if Ks==1; subplot ('position', [.065 .11 .7 .3439]);
else; subplot (212); end;
for i=1: m;
    plot ( [0 0] +fn (i), [-40 yn (i)],'b'); hold on;
end;
xlabel ( ['f (' fstr ')']); ylabel ('dB'); v=axis; zoom xon;
dv= (v (2) -v (1)) *.05; axis ( [v (1) -dv v (2) +dv v (3: 4)]); hold off;
if Kw==1;
    if Kmth==1; m=dF/f; elseif Kmth==2; m=K; end;
```

```
        mstr = ['m = ' num2str(m)];
        text(v(1) - dv, -5, mstr,'color','r','fonta','italic');
    end;
    set(gcf,'num','off','Name',[astr blanks(10) 'BBI 2000']);
```

fmpm 函数的运行结果如图 6.7 所示，为了清楚地表明频率的变化，缺省的调制信号频率设定为 2kHz，载波信号频率设定为 20kHz，调频的频偏设定为 10kHz。当调制信号为正弦波时，在频谱图形中标出了具体的调制指数 m。仿真结果清楚地表明，角度调制的频带宽度要大于幅度调制的频带宽度。

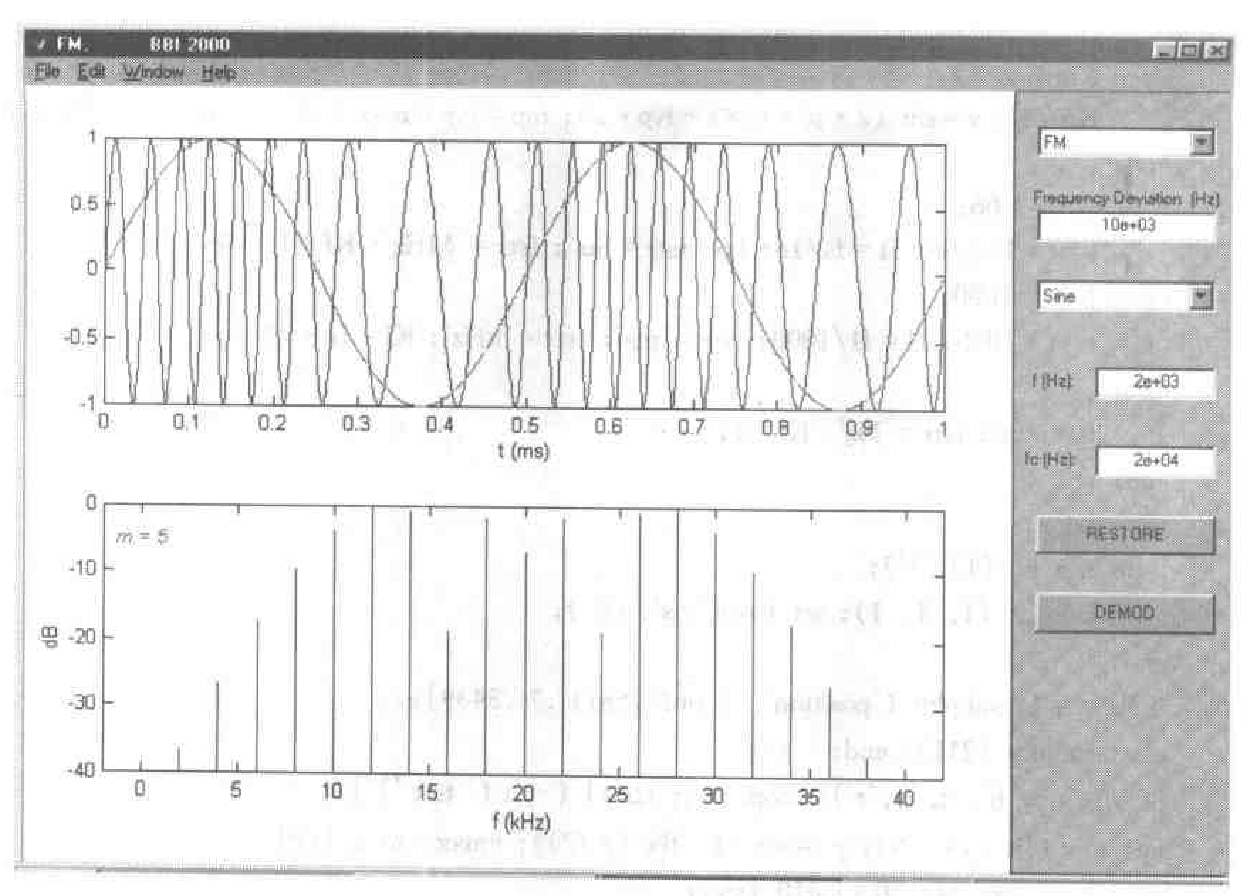


图 6.7 角度调制与解调

6.4 调频立体声广播

调频立体声广播是一种频分多路传输方式（FDM），它与普通的单声道调频广播是兼容的。调频立体声广播采用和差方式来传送立体声节目，“和信号”是左声道和右声道的信号之和，而“差信号”则是左声道和右声道信号之差。

调频立体声广播的基带信号 $m_b(t)$ 由下式来描述：

$$m_b(t) = (L + R) + (L - R) \cdot \cos\omega_{s}t + P \cdot \cos\frac{\omega_s}{2}t$$

的角频率，P 代表导频信号。

和信号为主信道，其频率范围是50Hz－15kHz，采用普通的调频方式进行传送；差信号先采用 38kHz 的副载波进行双边带调幅（SSB－SC），这称为副信道，副信道的频率范围23kHz－53kHz。为了便于在调频接收机内进行同步解调，在基带信号中还设置了导频信号，其频率为19kHz，为副载波频率的一半。对基带信号进行频率调制（频偏为75kHz），这样就形成了立体声广播的射频信号。

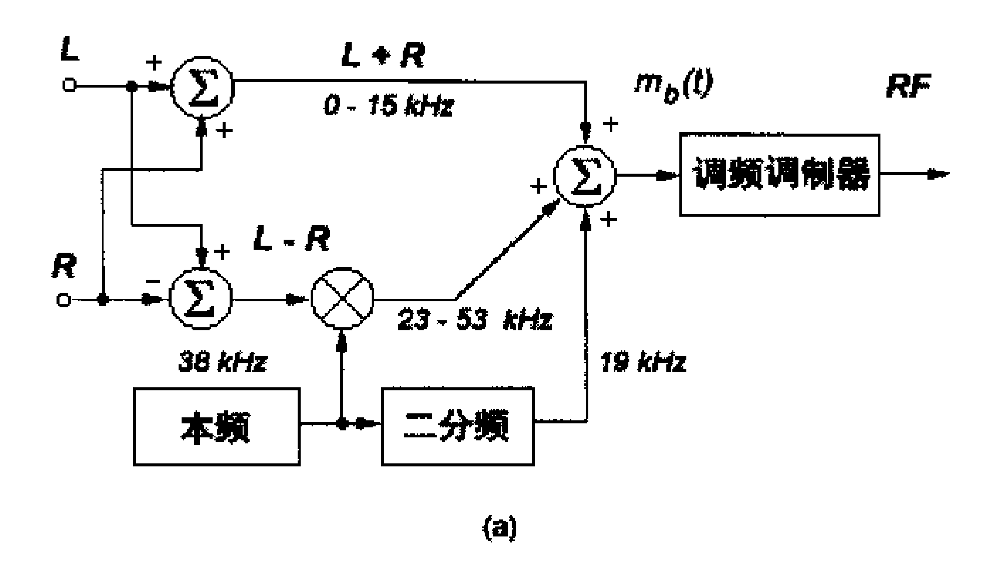
在调频立体声接收机中，在变频和中放之后首先进行调频解调；然后使用锁相环电路，根据导频信号来恢复副载波，这样确保了副载波的频率与相位与发射端严格同步；接着通过一个低通滤波器和一个高通滤波器将主信道信号和副信道信号分开，并对副信道进行同步解调，这样就得到了和信号和差信号；最后通过矩阵电路将和信号和差信号还原为左声道信号和右声道信号。

调频立体声广播的系统框图如图 6.8 所示，其中（a）图为发射机的系统框图，（b）图为接收机的系统框图。

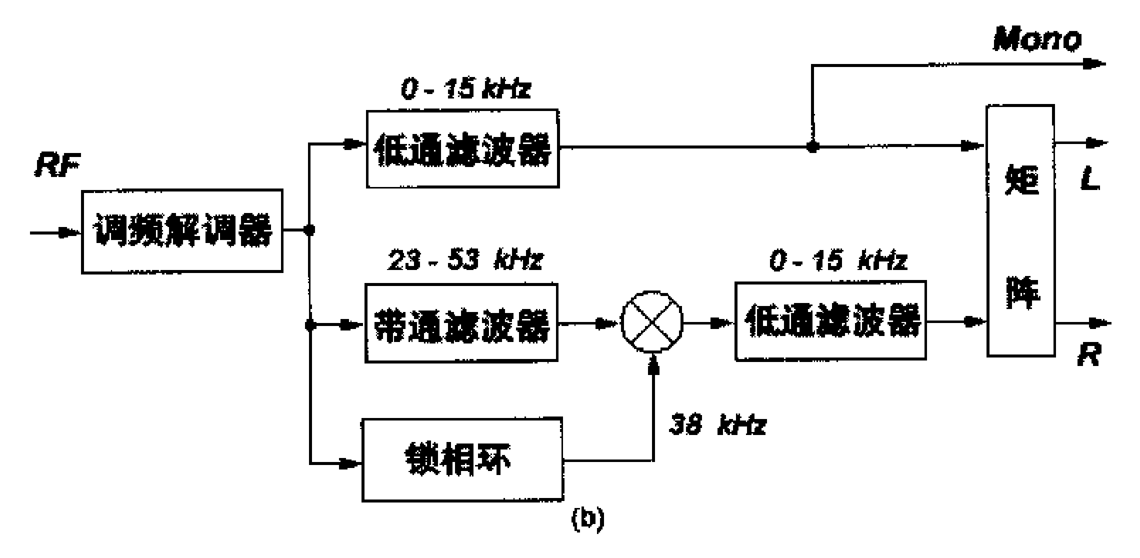
自定义的 M 函数 fmstereo.m 用来对调频立体声广播系统进行仿真，它在 4.2 版和 5.x 版内均可以运行。为了便于使用，在函数中采用了用户控制框来输入各种参数。输入的左右声道信号可以在正弦波、方波和锯齿波之间进行选择，信号的频率也可以进行指定，缺省的频率为 1kHz；输入信号选择完毕之后，按 START 按钮便可以进行仿真，运行结果可以显示基带信号的波形和频谱、已调波信号的波形和频谱、解调后的基带信号的波形和频谱、输出信号的波形和频谱。

fmstereo.m 函数的程序清单如下：

```
function [] =fmstereo (action);
%
% BBI 2000
global L R yb yrf y1 LL RR fl fr fc fs fg f1 t v N H12 H14 H22 H24...
          Kenv H5 H6 H7 H8
v=version; if nargin<1; action='initialized'; end;
Kact=0;
if strcmp (action,'initialized');
     if strcmp (v (1),'5');
          figure ('Num','off','Units','pix','pos', [5 29 792 530], ...
               'Color', [1 1 1]);
elseif strcmp (v (1),'4');
     figure ('num','off','units','pix','pos', [5 29 792 530]);
     whitebg ('w');
end;
set (gcf,'name', ...
     ['FM Stereo System ( Subcarrier 38 kHz, Pilot Tone 19 kHz,'...
     ' Frequency Deviation 75 kHz ), BBI 2000']);
```



(a)



(b)

图 6.8　导频制调频立体声广播系统

```
uicontrol ('Style','Frame','Units','Normal','Pos', [.82 0 .2 1], ...
          'Back', [.8 .8 .8]);

H11 = uicontrol ('Style','Text','Units','Normal','Pos', ...
                 [.84 .95 .15 .03],'String','Left Channel', ...
                 'Back', [.8 .8 .8],'fore','b');

H12 = uicontrol ('Style','Popup','Units','Normal','Pos', ...
               [.84 .9 .15 .05], ...
               'String', str2mat ('Sine','Square','Sawtooth'), ...
               'Back', [1 1 1],'Callback','fmstereo left;');
```

```
H13 = uicontrol ('Style','Text','Units','Normal','Pos', ...
        [.83 .835 .05 .04],'String','f (Hz):','Back', [.8 .8 .8]);

H14 = uicontrol ('Style','Edit','Units','Normal','Pos', ...
        [.89 .84 .1 .04],'Back', [1 1 1],'String', ...
        '1e+03','Call','fmstereo fl;');

H21 = uicontrol ('Style','Text','Units','Normal','Position', ...
            [.84 .75 .15 .03],'String','Right Channel', ...
            'Back', [.8 .8 .8],'fore','r');

H22 = uicontrol ('Style','Popup','Units','Normal','Position', ...
      [.84 .7 .15 .05],'String', str2mat ('Sawtooth','Sine','Square'), ...
            'Back', [1 1 1],'Callback','fmstereo right;');

H23 = uicontrol ('Style','Text','Units','Normal','Pos', ...
        [.83 .635 .05 .04],'String','f (Hz):','Back', [.8 .8 .8]);

H24 = uicontrol ('Style','Edit','Units','Normal','Position', ...
        [.89 .64 .1 .04],'Back', [1 1 1],'String','1e+03', ...
        'Call','fmstereo fr;');

H3 = uicontrol ('Style','Push','Units','Normal','Position', ...
       [.84 .55 .15 .05],'String','START','Call','fmstereo start;');

H41 = uicontrol ('Style','Text','Units','Normal','Position', ...
       [.822 .475 .06 .04],'String','fc (Hz):','Back', [.8 .8 .8]);

H42 = uicontrol ('Style','Edit','Units','Normal','Position', ...
            [.89 .48 .1 .04],'Back', [1 1 1], ...
            'String','4.5e+06','Call','fmstereo fc;');

H5 = uicontrol ('Style','Push','Units','Normal','Position', ...
           [.84 .4 .15 .05],'String','BASEBAND', ...
           'Call','fmstereo basehand;','vis','off');

H6 = uicontrol ('Style','Push','Units','Normal','Position', ...
           [.84 .32 .15 .05],'String','FM SIGNAL', ...
```

```
            'Call','fmstereo fm;','vis','off');

    H7=uicontrol ('Style','Push','Units','Normal','Position', ...
            [.84 .24 .15 .05],'String','DEMOD', ...
            'Call','fmstereo demod;','vis','off');
    H8=uicontrol ('Style','Push','Units','Normal','Position', ...
            [.84 .16 .15 .05],'String','OUTPUT SINGAL', ...
            'Call','fmstereo output;','vis','off');
    fc=4.5e+06; fl=1000; fr=fl; fg=38e+03; fs=4*fc;
    dt=1/fs; T=2/fr; t=0: dt: T-dt; f1=1/T; N=length (t); Kact=1; Kenv=1;

elseif strcmp (action,'left') |strcmp (action,'right');
    Kact=1;

elseif strcmp (action,'fl') |strcmp (action,'fr');
    fl=get (H14,'string');          fl=str2num (fl);
    fr=get (H24,'string');          fr=str2num (fr);
    f=min ( [fl fr]);               Kact=1;
    dt=1/fs;         T=2/f;         t=0: dt: T-dt; f1=1/T; N=length (t);

elseif strcmp (action,'fc');
    fc=get (gco,'string'); fc=str2num (fc); Kact=1;

elseif strcmp (action,'start');
    y=L+R+.01*sin (pi*fg*t) + (L-R) .*sin (2*pi*fg*t);
    yb=y/max (abs (y));                                          %基带
    yrf=sin (2*pi*fc*t+2*pi* (75e+03) *cumsum (yh) /fs);         % 射频
    y1=demod (yrf, fc, fs,'fm');                                 % 解调信号
    [b, a] =cheby2 (4, 40, .5*fc/fs/2); y1=filter (b, a, y1);
    if strcmp (v (1),'4');
        subplot (1, 1, 1); set (gca,'vis','off');
    end;
    k1=1.6; [h, a] =cheby2 (4, 40, k1*fg/fs);      u=filter (b, a, y1);
    [b, a] =cheby2 (4, 40, fg/fs/k1,'high');       v=filter (b, a, y1);
    v=demod (v, fg, fs,'amssb') *2;                LL=u+v; RR=u-v;
    subplot ('position', [.065 .5811 .7 .3439]);
    plot (t, L,'b', t, R,'r', [t (1) t (N)], [0 0],'k:'); zoom xon;
    xlabel ( ['t (s)']); title ('Input Singal');
    suhplot ('position', [.065 .11 .7 .3439]);
```

```
    plot (t, LL,'b', t, RR,'r', [t (1) t (N)], [0 0],'k:'); zoom xon;
    xlabel ( ['t (s)']); title ('Output Singal');
    set (H5,'vis','on'); set (H6,'vis','on');
    set (H7,'vis','on'); set (H8,'vis','on');

elseif strcmp (action,'baseband');
    fmst (t, yb, v, f1, N,'Baseband', Kenv); Kenv= -Kenv;

elseif strcmp (action,'fm');
    fmst (t, yrf, v, f1, N,'FM');

elseif strcmp (action,'demod');
    fmst (t, y1, v, f1, N,'Demodulated');

elseif strcmp (action,'output');
    if strcmp (v (1),'4');
        subplot (1, 1, 1); set (gca,'vis','off');
    end;
    subplot ('position', [.065 .5811 .7 .3439]);
    plot (t, L,'b', t, R,'r', [t (1) t (N)], [0 0],'k:'); zoom xon;
    xlabel ( ['t (s)']); title ('Input Singal');
    subplot ('position', [.065 .11 .7 .3439]);
    plot (t, LL,'b', t, RR,'r', [t (1) t (N)], [0 0],'k:'); zoom xon;
    xlabel ( ['t (s)']); title ('Output Singal');

end;

if Kact= =1;
    if strcmp (v (1),'4');
        subplot (1, 1, 1); set (gca,'vis','off');
    end;
    k=get (H12,'value');
    if k= =1; L=sin (2 * pi * fl * t+pi/2);
    elseif k= =2; L=square (2 * pi * fl * t, 50);
    elseif k= =3; L=sawtooth (2 * pi * fl * t+pi/6, .5);
    end;
    k=get (H22,'value');
    if k= =2; R=sin (2 * pi * fr * t+pi/3);
    elseif k= =3; R=square (2 * pi * fr * t-pi/6, 50);
```

```
    elseif k= =1; R=sawtooth (2*pi*fr*t, .5);
    end;
    subplot ('position', [.065 .5811 .7 .3439]); plot (t, L, [t (1) t (N)], [0 0],'
k:');
    xlabel ( ['t (s)']); title ('Lelf Channel');
    subplot ('position', [.065 .11 .7 .3439]);
    plot (t, R,'r', [t (1) t (N)], [0 0],'k:');
    xlabel ( ['t (s)']); title ('Right Channel');
end;
```

fmstereo.m 函数在运行过程中要调用 fmst.m 函数，该函数清单如下：

```
function [] =fmst (t, y, v, f1, N, tstr, Kenv);
%
global L R
if nargin<7; Kenv=0; end;

if strcmp (v (1),'4');
     subplot (1, 1, 1); set (gca,'vis','off');
end;
suhplot ('position', [.065 .5811 .7 .3439]);
if Kenv= =1;
   plot (t, y,'b', t, L,'b:', t, R,'r:');
else;
   plot (t, y, [t (1) t (N)], [0 0],'k:');
end; zoom xon;
xlabel ( ['t (s)']); title ( [tstr ' Signal']);
yy=abs (fft (y)) /N; yy=yy (1: fix (N/2)); ymax=max (yy);
yy=yy/ymax; yy=20*log10 (yy);
I=find (yy>-50); fn= (I-1) *f1; yn=yy (I); m=length (I);
subplot ('position', [.065 .11 .7 .3439]);
for i=1: m;
   plot ( [0 0] +fn (i), [-60 yn (i)],'b'); hold on;
end;
xlabel ( ['f (Hz)']); ylabel ('dB'); v=axis; zoom xon;
dv= (v (2) -v (1)) *.05; axis ( [v (1) -dv v (2) +dv v (3: 4)]);
title ('Spectrum'); hold off;
if Kenv~ =1;
     text (19e+03, -30,'Pilot Tone','fontn','arial', ...
```

```
        'fonts', 9,'color','b','hor','center');
end;
```

图6.9为fmstereo函数的运行结果。在这个实例中，左声道信号为正弦波，频率为2000Hz，右声道信号为锯齿波，频率为1000Hz，从中可以清楚地看出基带信号的包络分别代表着左声道信号和右声道信号，这一点是调频立体声广播系统特有的。

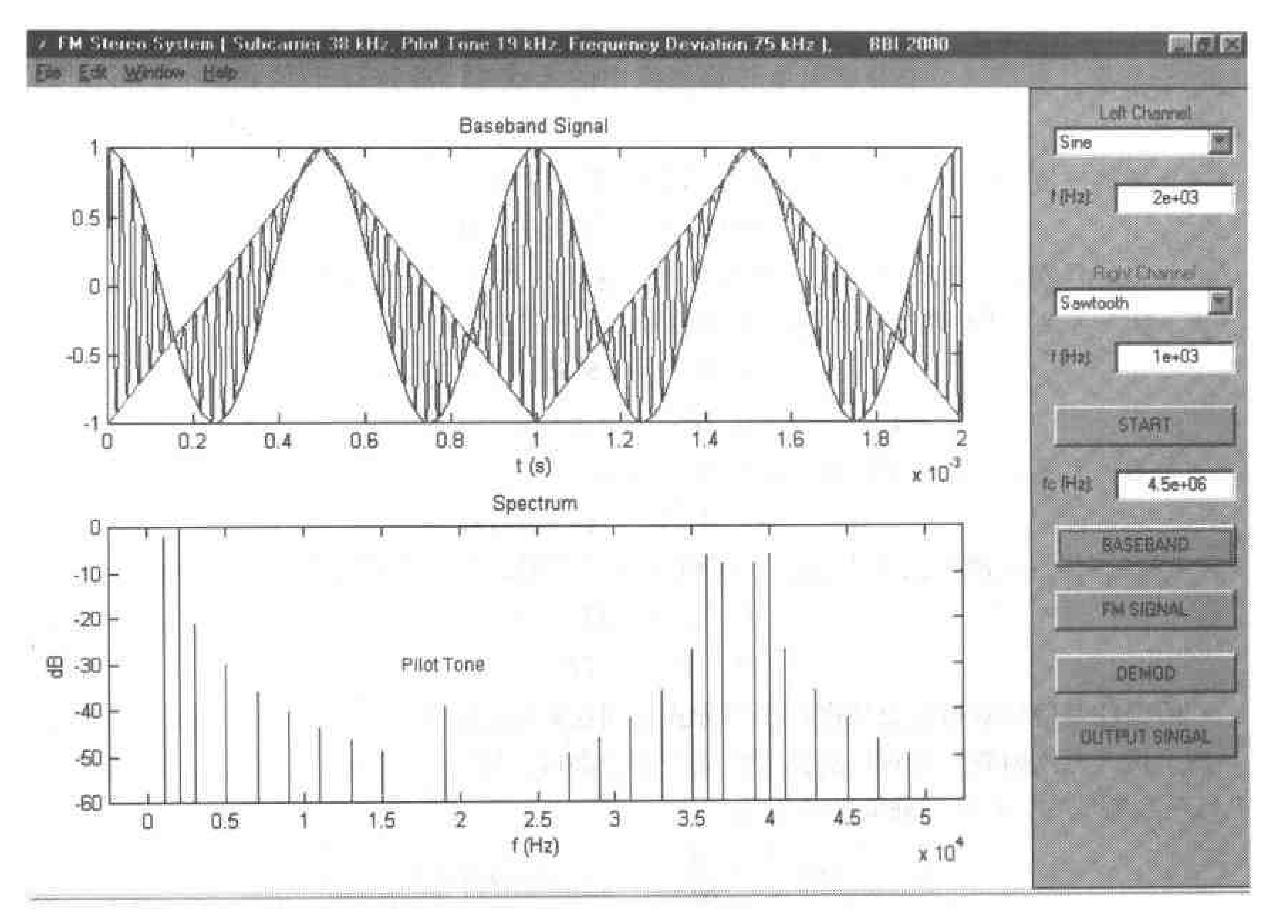


图6.9　调频立体声广播

6.5　电视信号发生器

自定义M函数videos.m本身相当于一个电视的行信号发生器，它调用若干M函数来对PAL－D制式的彩色电视信号进行仿真，其主要功能是生成各种标准测试信号，显示信号的波形和频谱，对电视信号进行解码后，分别显示三基色信号、同时显示该信号的电视图象。videos.m函数的功能丰富是一个相当实用的仿真软件。

彩色全电视信号由亮度信号、色差信号（色度信号）、色同步信号、行同步信号、行消隐信号、场同步信号和场消隐信号组成，由于PAL制的场频为50Hz，扫描一场的时间为20ms，因此要仿真一场的电视信号所需的数据量太大，故难以实现；所幸的是大多数的测试信号均为行信号，即在一场中各行的信号是完全相同的，因此我们只需对行信号进行仿真

就可以了。常用的行测试信号有：2T 正弦平方波信号 B1 和条脉冲信号 B3、多波群信号 C、阶梯波信号 D1、阶梯波叠加副载波信号 D2、250kHz 方波信号 E、副载波填充的 10T 信号 F 和副载波填充的条脉冲信号 G、10T 调制的副载波信号 F1 和条脉冲调制的副载波信号 G1、三电平色度信号 G2 等等，这些在 videos.m 函数中均有体现。

在 PAL 制中，行频为 15625Hz，扫描一行的时间为 64μs，其中行正程为 52μs，行同步信号宽度为 4.7μs，行同步电平为 -0.3V，行消隐电平为 0V，白电平为 0.7V。

实现与黑白电视的兼容，彩色电视广播并不是直接传送红绿蓝三基色信号，而是传送所谓的传输三基色：一个亮度信号和两个色差信号。彩色摄像机输出三基色信号 E_R，E_G，E_B，它们都是电压，为了补充光电转换过程中的非线性，要进行 γ 校正，校正后就成为 E_R'，E_G'，E_B'信号。

亮度信号 Y 可由亮度方程来表示，表示成电压后为，

$$E_Y' = 0.299E_R' + 0.587E_G' + 0.114E_B'$$

色差信号就是基色信号与亮度信号之差，这样一共有三个色差信号：R-Y，B-Y，G-Y。彩色电视广播中需要传输两个色差信号 R-Y 和 B-Y。

$$E_{R-Y} = 0.7E_R' - 0.59E_G' - 0.11E_B'$$

$$E_{B-Y}' = -0.3E_R' - 0.59E_G' + 0.89E_B'$$

而 G-Y 信号可以从另外两个色差信号求出，

$$E_{G-Y}' = -0.51E_{R-Y}' - 0.19E_{G-Y}'$$

PAL 制传送的两个色差信号称为 U 信号和 V 信号，它们的带宽均为 1.3MHz。

$$E_U' = 0.493E_{B-Y}'$$

$$E_V' = 0.877E_{R-Y}'$$

两个色差信号采用正交平衡调幅的方式调制在彩色副载波上，为了近可能减少对亮度信号的干扰，彩色副载波的频率选择在视频带宽的高端，同时彩色副载波是与行频锁定的。在 PAL-D 制式中，彩色副载波的频率为，

$$f_{sc} = \left(283\frac{3}{4} + \frac{1}{625}\right) \cdot f_h = 4.43361875\ MHz$$

色差信号对彩色副载波调制之后形成的色度信号为，

$$e_c(t) = \begin{cases} E_U' \cdot \sin\omega_{sc}t + E_V' \cdot \cos\omega_{sc}t & (N) \\ E_U' \cdot \sin\omega_{sc}t - E_V' \cdot \cos\omega_{sc}t & (P) \end{cases}$$

从上式可以看出逐行倒相的过程。一行不倒相，称为 N 行，N 行与 NTSC 制非常相近；另一行倒相，称为 P 行。

PAL 制色同步信号的表达式为，

$$e_b(t) = \begin{cases} 0.3\sin(\omega_{sc}t + 135^\circ) \\ 0.3\sin(\omega_{sc}t + 225^\circ) \end{cases}$$

色同步信号位于行同步信号的后面，持续时间为 2.26μs。亮度信号、色度信号、色同步信号、同步信号和消隐信号加在一起就构成了完整的 PAL 制彩色电视信号。

自定义的 M 函数 videos.m 在 4.2 版和 5.x 版内均可以运行，用来确定一些参数。函数的程序清单如下：

```
function [t, s, kn, sam] =videos (fs, fc);
%
% CHE QING, BBI 96/11
if nargin<1; fs=3*4.43361875; end;                    % 缺省的采样频率
if nargin<2; fc=fs*.5; end;                           % 缺省的载波频率
global K ksmth ksmth1 nback pb0;
dt=1/fs; k1=fix (4.7/dt) +1; kn=fix (64/dt);
i=1: kn; t= (i-1) *dt; tk=t (k1); i=1: k1;            % 行同步脉冲
s (i) =i*0-.3; i=k1+1: kn; s (i) =0*i;
nb=round (9/dt); nb=1: nb;
t= [t t (nb) + (kn+1) *dt]; s= [s s (nb)];
b75= [1 1 1; .75 .75 0; 0 .75 .75; 0 .75 0; ...
.75 0 .75; .75 0 0; 0 0 .75; 0 0 0];
b100= [1 1 1; 1 1 0; 0 1 1; 0 1 0; 1 0 1; 1 0 0; 0 0 1];
K=1; ksmth=1; ksmth1=1; nback=1;
back=str2mat ('White','Yellow','Cyan','Green','Magenta','Red','Blue');
save b_pulse fs fc dt kn tk t s b75 back b100;        % 存储数据文件
[t, s, H] =videos1;
```

videos.m 函数直接调用 videos1.m 函数，该函数用来建立一个用户控制框，并生成各种行测试信号，其程序清单如下：

```
function [t, s, H, tk, sam] =videos1 (action, cstr);
%
%     BBI      96/11
if nargin<2; cstr='g'; end;
if nargin<1; action='initialize'; end;

global K ksmth vd aa k_sig rgb c cmap ca ksmth1 nback phis...
actionLast figNumber;

load h_pulse;
% ======================================    %生成色同步信号
fsc=4.43361875; T=1/fsc;
k2=fix (25*T/dt+1e-06) +1; k3=fix ( (35*T) /dt);
w=2*pi*fsc;    ph=pi-K*pi/4;    H= [];    ph0=rand (1) *2*pi;
for i=k2: k3;
   s (i) =.15*sin (w* (i-1) *dt+ph+ph0);
```

```
end;
ph1 = pi + K * pi/4;
for i = k2: k3;
    s (i + kn) = .15 * sin (w * (i - 1 + kn) * dt + ph1 + ph0);
end;
tk = [tk t (k2)];
% ==========================================================
if strcmp (action, 'initialize'),
    figNumber = figure ('Name', 'VIDEO TEST SIGNAL GENERATOR BBI', ...
        'Num', 'off', 'pos', [5 29 792 530]);
      axes ('Units', 'normalized', 'Pos', [0.1 0.1 0.7 0.84]);
      whitehg ('k'); set (gcf, 'color', [0 0 0]);

      % =================================
      % Information for all huttons
      labelColor = [0.8 0.8 0.8];
      top = 0.975;
      left = 0.84;
      bottom = 0.025;
      btnWid = 0.15;
      btnHt = 0.05;
      spacing = 0.01;
      % ====================================
      % The CONSOLE frame
      frmBorder = 0.01;
      yPos = 0.02 - frmBorder;
      frmPos = [left - frmBorder yPos btnWid + 2 * frmBorder 0.94 + 4 * frmBorder];
      cslHndl = uicontrol ('Style', 'frame', 'Units', 'normalized', ...
          'Position', frmPos, 'Back', [0.20 0.40 0.40]);
      set (gcf, 'UserData', cslHndl);
      % ====================================
      % The H - pulse button
      btnNumber = 1;
      yPos = top - (btnNumber - 1) * (btnHt + spacing);
      labelStr = 'H - pulse';
      callbackStr = ' [t, s, H, tk] = videos1 (''H - pulse'');';
      btnPos = [left yPos - btnHt btnWid btnHt];
      uicontrol ('Style', 'pushbutton', 'Units', 'normalized', ...
          'Pos', btnPos, 'String', labelStr, 'Callback', callbackStr);
```

```
% =====================================
% The K Signal button
btnNumber=2;
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='K Signal';
callbackStr=' [t, s, H, tk] =videos1 (''K Signal'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% =====================================
% The B1-B3 Signal button
btnNumber=3;
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='B1 and B3';
callbackStr=' [t, s, H, tk] =videos1 (''B1-B3'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% =====================================
% The D1 Signal button
btnNumber=4;
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='D1 Signal';
callbackStr=' [t, s, H, tk, sam] =videos1 (''D1'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% =====================================
% The D2 Signal button
btnNumber=5;
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='D2 Signal';
callbackStr=' [t, s, H, tk] =videos1 (''D2'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% =====================================
% Tbe C Signal button
btnNumber=6;
```

```
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='C Signal';
callbackStr=' [t, s, H, tk] =videos1 (''C'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% ======================================
% The E Signal button
btnNumber=7;
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='E Signal';
callbackStr=' [t, s, H, tk] =videos1 (''E'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);

% ======================================
% The F & G Signal button
btnNumber=8;
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='F and G';
callbackStr=' [t, s, H, tk] =videos1 (''F&G'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% ======================================
% The 331 line button
btnNumber=9;
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='331 line';
callbackStr=' [t, s, H, tk] =videos1 (''331L'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);

% ======================================
% The COLOR BAR SIGNAL button
btnNumber=10;
yPos=top- (btnNumber-1) * (btnHt+spacing);
```

```
labelStr='BAR';
callbackStr=' [t, s, H, tk] =videos1 (''BAR'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% ==========================================
% The COLOR BACK SIGNAL hutton
btnNumber=11;
yPos=top- (btnNumber-1) * (btnHt+spacing);
labelStr='Color Back';
callbackStr=' [t, s, H, tk] =videos1 (''BACK'');';
btnPos= [left yPos-btnHt btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callhackStr);
% ==========================================
% The SPECTRUM button
labelStr='Spectrum';
callbackStr=' [t, s, H, tk] =videos1 (''SPECTRUM'');';
btnPos= [left bottom+btnHt*3+spacing*3 btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% ==========================================
% The Monitor button
labelStr='MONITOR';
callbackStr=' [t, s, H, tk] =videos1 (''MONITOR'');';
btnPos= [left bottom+btnHt*2+spacing*2 btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbackStr);
% ==========================================
% The RGB/BW button
labelStr='RGB/BW';
callbackStr=' [t, s, H, tk] =videos1 (''RGB/BW'');';
btnPos= [left bottom+btnHt+spacing btnWid btnHt];
uicontrol ('Style','pushbutton','Units','normalized', ...
    'Pos', btnPos,'String', labelStr,'Callback', callbeckStr);
% ==========================================
% The INFO button
uicontrol ('Style','push','Units','normalized', ...
    'Pos', [left bottom btnWid btnHt],'String','Info', ...
```

```
            'Callback','videos1 (''info'');');

        % ===================================
        a='H-pulse'; k_sig=-1;

elseif strcmp (action,'K Signal'),
        k2=fix (12/dt) +1; k3=fix (62/dt) +1; tk= [tk t (k2) t (k3)];
        i=k2: k3; s (i) =i*0+.35; a='K Signal'; k_sig=-1;

elseif strcmp (action,'H-pulse'),
        a='H-pulse'; k_sig=-1;

elseif strcmp (action,'B1-B3'),
        fc=6; T=1/ (2*fc);
        k2=ceil ( (17-T*2) /dt) +1; k3=floor ( (17+T*2) /dt) +1;
        k4=fix (32/dt) +1; k5=fix (57/dt) +1;
        tk= [tk t (fix (17/dt) +1) t (k4) t (k5)]; ts=fix (17/dt) *dt;
        for i=k2: k3;
          s (i) =.35* (1-cos (6*pi* ( (i-1) *dt-ts) -pi));
        end;
        for i=k4: k5; s (i) =.7; end;
        a='B1 & B3 Signal'; k_sig=-1;

elseif strcmp (action,'D1');
        T=52/6;
        k2=fix ( (10.5+T) /dt) +1;        k3=fix ( (10.5+2*T) /dt) +1;
        k4=fix ( (10.5+3*T) /dt) +1;      k5=fix ( (10.5+4*T) /dt) +1;
        k6=fix ( (10.5+5*T) /dt) +1;      k7=fix ( (10.5+6*T) /dt) +1;
        for i=k2: k3-1;   s (i) =s (i) +.14;   end;
        for i=k3: k4-1;   s (i) =s (i) +.28;   end;
        for i=k4: k5-1;   s (i) =s (i) +.42;   end;
        for i=k5: k6-1;   s (i) =s (i) +.56;   end;
        for i=k6: k7;     s (i) =s (i) +.70;   end;
        tk= [tk t (k2) t (k3) t (k4) t (k5) t (k6) t (k7)];
        a='D1 Signal'; k_sig=-1;
        m=.875;                               %   调制度   87.5%
        sam= (m*abs (s-.7) +1-m) .*cos (2*pi*fc*t);
elseif strcmp (action,'D2');
        T=52/6; w=2*pi*fsc;
```

```
        k1=fix (10.5/dt) +1;                    k2=fix ( (10.5+T) /dt) +1;
        k3=fix ( (10.5+2*T) /dt) +1;            k4=fix ( (10.5+3*T) /dt) +1;
        k5=fix ( (10.5+4*T) /dt) +1;            k6=fix ( (10.5+5*T) /dt) +1;
        k7=fix ( (10.5+6*T) /dt) +1;
        for i=k1: k2-1;
            s (i) =s (i) +.14*sin (w* (i-1) *dt+ph0);
        end;
        for i=k2: k3-1;
            s (i) =s (i) +.14+.14*sin (w* (i-1) *dt+ph0);
        end;
        for i=k3: k4-1;
            s (i) =s (i) +.28+.14*sin (w* (i-1) *dt+ph0);
        end;
        for i=k4: k5-1;
            s (i) =s (i) +.42+.14*sin (w* (i-1) *dt+ph0);
        end;
        for i=k5: k6-1;
            s (i) =s (i) +.56+.14*sin (w* (i-1) *dt+ph0);
        end;
        for i=k6: k7;
            s (i) =s (i) +.70+.14*sin (w* (i-1) *dt+ph0);
        end;
        tk= [tk t (k2) t (k3) t (k4) t (k5) t (k6) t (k7)];
        a='D2 Signal'; k_sig=-1;

elseif strcmp (action,'C'),
        k2=fix (12/dt) +1;          k3=fix (16/dt) +1;
        k4=fix (20/dt) +1;          k5=fix (62/dt) +1;
        tk= [tk t (k2) t (k4) t (k5)];
        i=k2: k3;                   s (i) =i*0+.7;
        i=k4: k5;                   s (i) =i*0+.35;
        w= [0.5 1.5 2.5 4 4.8 5.8] *2*pi;
        for n=1: 6;
          ts=n*6+18.02;
          k1=fix (ts/dt) +1; k2=fix ( (ts+4.5) /dt) +1; tk= [tk t (k1)];
          for i=k1: k2;
              s (i) =s (i) +.21*cos (w (n) * (i-k1) *dt);
          end;
        end; a='C Signal'; k_sig=-1;
```

```
elseif strcmp (action,'E'),
   ts=10.5;
   for n=1: 13;
     k1=fix (ts/dt) +1; k2=fix ( (ts+2) /dt) +1;
     for i=k1: k2; s (i) =s (i) +.7; end;
     ts=ts+4;
   end; a='E Signal'; k_sig=-1;
elseif strcmp (action,'F&G'),
     fc=6; T=1/ (2*fc);
     k2=ceil ( (17-T*10) /dt) +1; k3=floor ( (17+T*10) /dt) +1;
     k4=fix (32/dt) +1; k5=fix (57/dt) +1;
     tk= [tk t (fix (17/dt) +1) t (k4) t (k5)];
     ts=fix (17/dt) *dt;
     for i=k2: k3;
         s (i) =.35* (1-cos (1.2*pi* ( (i-1) *dt-ts) -pi));
         s (i) =.5*s (i) .* (1+cos (w* ( (i-1) *dt-ts+ph0)));
     end;
     for i=k4: k5;
         s (i) =.35* (1+cos (w* (i-k4) *dt+ph0));
     end;       a='F & G Signal';       k_sig=-1;
elseif strcmp (action,'331L');
       k1=ceil (12/dt) +1; k2=floor (62.5/dt) +1;
       i=k1: k2;                  s (i) =i*0+.35;
       k1=ceil (14/dt) +1; k2=ceil (18/dt) +1;
       k3=ceil (22/dt) +1; k4=floor (28/dt) +1;
       tk= [tk t (k1) t (k2) t (k3) t (k4)];
       for i=k1: k2-1;
           s (i) =s (i) +.07*cos (w* (i-k1) *dt+ph0);
       end;
       for i=k2: k3-1;
           s (i) =s (i) +.21*cos (w* (i-k1) *dt+ph0);
       end;
       for i=k3: k4;
           s (i) =s (i) +.35*cos (w* (i-k1) *dt+ph0);
       end;
       k1=ceil (32/dt) +1; k2=floor (60/dt) +1; tk= [tk t (k1) t (k2)];
       for i=k1: k2;
           s (i) =s (i) +.21*cos (w* (i-k1) *dt+ph0);
```

```
    end; a='331 line'; k_sig=-1;
elseif strcmp (action,'BAR'),
  ts=10.5; w=2*pi*fsc; xx=0; yy=0;
  for n=1: 8;                                          %    yc= [y u v]
    yc=vdpal (b75 (n,:)) *.7;
    k1=fix (ts/dt) +1; k2=fix ( (ts+6.5) /dt);
    y=yc (1); u=yc (2); v=yc (3); c=sqrt (u*u+v*v);
    xx= [xx t (k1) t (k2)]; yy= [yy y y];
    for i=k1: k2;
        s (i) =s (i) +u*sin (w* (i-1) *dt+ph0) +K*v*cos (w* (i-1)
               *dt+ph0) +y;
        if s (i) >.7; s (i) =.7;
        elseif s (i) <-.175; s (i) =-.175;
        end;
    end;
    ts=ts+6.5;
  end; a='Bar Signal'; k_sig=-1;

elseif strcmp (action,'BACK')
    w=2*pi*fsc;
    yc=vdpal (b100 (nback,:)) *.7; % yc= [y u v]
    k1=fix (10.5/dt) +1; k2=fix ( (62.5) /dt);
    y=yc (1); u=yc (2); v=yc (3); c=sqrt (u*u+v*v);
    xx= [t (k1) t (k2)]; yy= [y y];
    for i=k1: k2
      s (i) =s (i) +u*sin (w* (i-1) *dt+ph0) +K*v*cos (w* (i-1) *
             dt+ph0) +y;
    end;
    a= [back (nback,:) ' Back Signal']; k_s   ig=-1;
    nback=nback+1;
    if nback>7;     nback=1;     end;

elseif strcmp (action,'SPECTRUM');
    n=kn; n2=floor (n/2);
    a=abs (fft (vd (1: n)) /n); a=20*log10 (a); amax=max (a);
    i=1: n2; f=1/64* (i-1);
    H=plot (f, a (i) -amax, cstr); hold on;
    plot ( [0 6], [-10 -10],'b:');
    plot ( [0 6], [-20 -20],'b:');
```

```
      plot ( [0 6], [-30 -30],'b:');
      plot ( [0 6], [-40 -40],'b:'); hold off;
      axis ( [0 6 -50 0]);
      xlabel ('Frequency ( MHz )');
      ylabel ('Relative Level ( dB )');
      title ('SPECTRUM ANALYZER');
      text (2.5, -3, aa);
end;
% ===============================================
if strcmp (action,'info') = =0 & strcmp (action,'SPECTRUM') = =0;
n=fix (10/dt); m1=kn-n+1: kn;
n=fix (16/dt); m2=round (9/dt): n;

if strcmp (action,'MONITOR');
   n=kn;                        as=abs (fft (vd (1: n)) /n);
   as=20 * log10 (as);          amax=max (as);
   as=as-amax;                  afc=max (as (250: 300));
   if k_sig<2;
     if afc>-17;
       [c, cmap, y, u, v, rgb, nq] =cdemod (t, vd, ksmth); ca=aa;
       if ksmth<0;
         ca= [ca ' ( ' 'smoothed ' ')'];
       end;
       ksmtb= -ksmth;
     else;
       [c, cmap] =ydemod (t, vd);
       ca=aa;
       end;
   end;
   linedisp (c, cmap, ca); k_sig=1;
 elseif strcmp (action,'RGB/BW');
   n=kn;                        as=abs (fft (vd (1: n)) /n);
   as=20 * log10 (as);          amax=max (as);
   as=as-amax;                  afc=max (as (250: 300));
   if k_sig=1;
       if afc>-17;
       [c, cmap, y, u, v, rgb, nq] =cdemod (t, vd, ksmth1); ca=aa;
       if ksmth1<0;
         ca= [ca ' ( ' 'smoothed ' ')'];
```

```
        end;
        ksmth1 = -ksmth1;
      else;
        [c, cmap] =ydemod (t, vd); ca=aa;
      end;
    end;
    if afc> -17;
      plot ( [0 2000], [0 0],'w:', [0 2000], [1 1],'w:'); hold on;
      rgbplot (rgb); hold off;
      axis ( [0 max ( [700 length (c)]) -.1 1.1]);
      title ('R G B Signals');
    else
      y= (c-1) /255;
      plot ( [0 2000], [0 0],'w:', [0 2000], [1 1],'w:'); hold on;
      plot (y,'m'); hold off;
      axis ( [0 max ( [700 length (c)]) -.1 1.1]);
      title ('Luminance Signal --- Y');
    end;
    k_sig=2;
  else;
    H=plot ( [-10 80], [0 0],'b:', [-10 80], [700 700],'b:', [-10 80], ...
          [-300 -300],'b:', 10.5, 0,'m+', 62.5, 0,'m+', ...
          [t (m1) -64 t t (m2) +64], [s (m1) s s (m2)] *1000, cstr);
    hold on;        vd=s;           aa=a;
    if K>0; ks='N line';
    else; ks='P line';
    end;
    axis ( [-10 80 -400 800]);
    text (35, -500, ['t (' 'μs)'],'HorizontalAlignment','center');
    ylabel ('Video Signal Level [ mV ]'); text (28, -270, a);
    m=length (tk); tk=floor ( (tk+.05) *10) *.1;
    text (tk (1) -1, -350, num2str (tk (1)));
    text (tk (2) -2, 180, num2str (tk (2)));
    text (7.9, -270, num2str (10.5)); text (57.9, -270, num2str (62.5));
    text (28, -340, ks); text (38, -340, [' ( ' num2str (ph/pi*180) ' )']);
    for i=3: m;
        ti=tk (i); text (ti-2, 830, num2str (ti));
    end; phis=ph0;
  end;
```

```
if strcmp (action,'C');
   text (24.8, 620,'0.5 1.5 2.5 4.0 4.8 5.8');
   text (23.0, 50, '40 120 200 320 380 460');
end;
if strcmp (action,'E');
   title ('250 kHz SQUARE WAVE ( 2 μs )');
end;
if strcmp (action,'BAR');
   plot (xx (4: 15), yy (4: 15) *1000,'m');
   title ('100 / 0 / 75 / 0');
   for i=4: 2: 14;
       text (61, yy (i) *1000, num2str (fix (yy (i) *1000+.5)));
   end;
end; K=-K;
end;      hold off;      actionLast=action;
```

videos1.m 函数在运行的过程中还要调用 vdpal.m 函数、cdemod.m 函数和 ydemod.m 函数。vdpal.m 函数的功能是根据 RGB 三基色来生成亮度和色度信号，cdemod.m 函数用来完成彩色信号的解调，而 ydemod.m 函数则用来完成黑白信号的解调。它们的程序清单如下：

```
function yuv=vdpal (c);
%
%      BBI 1997           亮度和色度信号
if nargin<1; c= [1 1 1]; end;

r=c (1); g=c (2); b=c (3);
y=.299*r+.587*g+.114*b;
u=.493* (b-y); v=.877* (r-y);
if abs (u) <1e-10; u=0; end;
if abs (v) <1e-10; v=0; end;
yuv= [y u v];
function [c, cmap, y, u, v, rgb, nq] =cdemod (t, s, ks, nbit)
%
% BB1   97/03/20          彩色信号解调
if nargin<5;    kn=851;    end;
if nargin<4;    nbit=7;    end;
if nargin<3; ks=1;         end;
global phis;
dt=t (2);
```

```
m1=fix (1/dt) +1;          m2=fix (4.8/dt) +1;        m3=fix (5.6/dt) +1;
n1=fix (10.5/dt);          n2=fix (62.5/dt);
black=s (m2);        s=s-black;        k=3/ (7*abs (s (m1)));
[b, a] =cheby2 (7, 50, .4,'high');     c=filtfilt (b, a, s (n1: n2));
k1=m3+3; k2=k1+1; w=4.43361875*2*pi; mbst=76: 102;
s1=s (mbst);     s2=s (mbst+kn);     s22= [s2 (2: 3) s2 (1: length (s2) -2)];
a1=fft (s1);     a2=fft (s22);
[tmp, i] =max (abs (a1));     ph1=angle (a1 (i));
Am=abs (a1 (i)) *2/length (s1);
[tmp, i] =max (abs (s2)); ph2=angle (s2 (i));
sinp=sin (ph2-ph1);          K=1;
if sinp>.8;          ph=pi*3/4;     K=1;          %     N line
elseif sinp<-.8;     ph=pi*5/4;     K=-1;         %     P line
end;
ss=s1 (1: 2);               ss=ss/Am;
A= [sin (w*t (76)) cos (w*t (76)); sin (w*t (77)) cos (w*t (77))];
phi=inv (A) *ss'; phs=asin (phi (2)); phc=acos (phi (1));
if phs<0;     ph0=2*pi-phc;
else;         ph0=phc;
end;
phi0=ph0-ph;
if phi0<0;     phi0=2*pi+phi0;     end;
[b1, a1] =cheby2 (4, 50, .6); sint=sin (w*t+phi0);
u0=2*k*s (n1: n2) .*sint (n1: n2);        %     u
u=filtfilt (b1, a1, u0);
cost=K*cos (w*t+phi0);
v0=2*k*s (n1: n2) .*cost (n1: n2);        %     v
v=filtfilt (b1, a1, v0);
[by, ay] =cheby2 (4, 50, .6);
y=k*filtfilt (by, ay, s (n1: n2));       %     y
r=v/.877+y;        b=u/.493+y; g= (y-.299*r-.114*b) /.587;
if ks<0; [r, g, b] =smth1 (y, r, g, b); end;        %  对彩色信号进行平滑
i=find (r>1); r (i) =ones (size (i));
i=find (r<0); r (i) =zeros (size (i));
i=find (g>1); g (i) =ones (size (i));
i=find (g<0); g (i) =zeros (size (i));
i=find (b>1); b (i) =ones (size (i));
i=find (b<0); b (i) =zeros (size (i));
nq=2^nbit-1;
```

```
r=round (r*nq) /nq;      g=round (g*nq) /nq;      b=round (b*nq) /nq;
map= [r' g' b'];          m=length (r);            rgh=map;
c=ones (1, m); k=1;       cmap= [];
neg= [-1 -1 -1];          v1=ones (m, 1);
for i=1: m;
   map1=map (i,:);
   if map1=neg;
     an=map- [map1 (1) *v1 map1 (2) *v1 map1 (3) *v1];
     an=an.*an; en=an (:, 1) +an (:, 2) +an (:, 3);
     N=find (en= =0);
     c (N) =k*ones (size (N)); cmap= [cmap; map1];
     map (N,:) =-1*ones (length (N), 3); k=k+1;
   end;
end;
```

```
function [y, ymap, sync, black] =ydemod (t, s)
%
%    BBI 1997      黑白信号解调
dt=t (2);
m1=fix (1/dt) +1;          m2=fix (4.8/dt) +1;
n1=fix (10.5/dt) +1;       n2=fix (62.5/dt);
sync=abs (s (m1)); black=s (m2); k=3/ (7*sync); s=s*k;
i=find (s>1);          s (i) =ones (size (i));
i=find (s<0);          s (i) =zeros (size (i));
y=fix (s (n1: n2) *255) +1; ymap=gray (256);
```

在彩色信号解调过程中色度信号与亮度信号的分离过程中使用了数字滤波器，而脉冲信号经过滤波后会产生“过冲”，因此引起彩色图象的边缘失真。在仿真的过冲之中使用平滑技术可以在很大的程度上消除“过冲”失真，smth1.m 函数就是用来对 RGB 信号进行平滑，其程序清单如下：

```
function [r, g, b, k, ky] =smth1 (y, r, g, b);
%
%    BBI 1997   三基色信号的平滑
ky=setfind (y); my=length (ky);
kr=setfind (r); mr=length (kr);
kb=setfind (b); mb=length (kb);
mk=max ( [my mr mb]);
```

```
if my= =mk; k=ky;
elseif mr= =mk; k=kr;
elseif mb= =mk; k=kb;
end;
if mk>1;
    k1=k (1); k2=k (2) -1; k12=k1: k2; n=length (k12);
    a=mean (r (k1+3: k2-3)); r (k12) =a*ones (1, n);
    a=mean (g (k1+3: k2-3)); g (k12) =a*ones (1, n);
    a=mean (b (k1+3: k2-3)); b (k12) =a*ones (1, n);
end;
if mk>2;
  for i=2: mk-1;
    k1=k (i); k2=k (i+1) -1; k12=k1: k2; n=length (k12);
    a=mean (r (k1+3: k2-3)); r (k12) =a*ones (1, n);
    a=mean (g (k1+3: k2-3)); g (k12) =a*ones (1, n);
    a=mean (b (k1+3: k2-3)); b (k12) =a*ones (1, n);
  end;
end;
if mk>0;
  k1=k (mk);          k2=length (r);
  if k2>k1+3
    k12=k1: k2;       n=length (k12);
    a=mean (r (k1: k2));        r (k12) =a*ones (1, n);
    a=mean (g (k1+3: k2));        g (k12) =a*ones (1, n);
    a=mean (h (k1+3: k2)); b (k12) =a*ones (1, n);
  end;
end;
```

图6.10、图6.11和图6.12分别给出了阶梯波测试信号D1的波形、频谱，以及该信号在电视屏幕上显示出的图像。由于D1信号为黑白信号，因此在彩色副载波处频谱的数值在-40dB以下。

图形界面上各种按钮的作用是很清楚的，其中RGB/BW按钮用来显示RGB信号或亮度信号，而MONITOR按钮用来显示信号所对应的电视图像。

6.6 彩色矢量示波器

彩色矢量示波器是用来显示彩色电视广播的色度信号，色度信号由色差信号R-Y和B-Y调制在彩色副载波上而构成的，它本身是一个矢量，其幅角代表色调，而其幅度表示色饱和度。

自定义的M函数vectorsc.m对彩色矢量示波器进行仿真，它可以方便地显示颜色与色

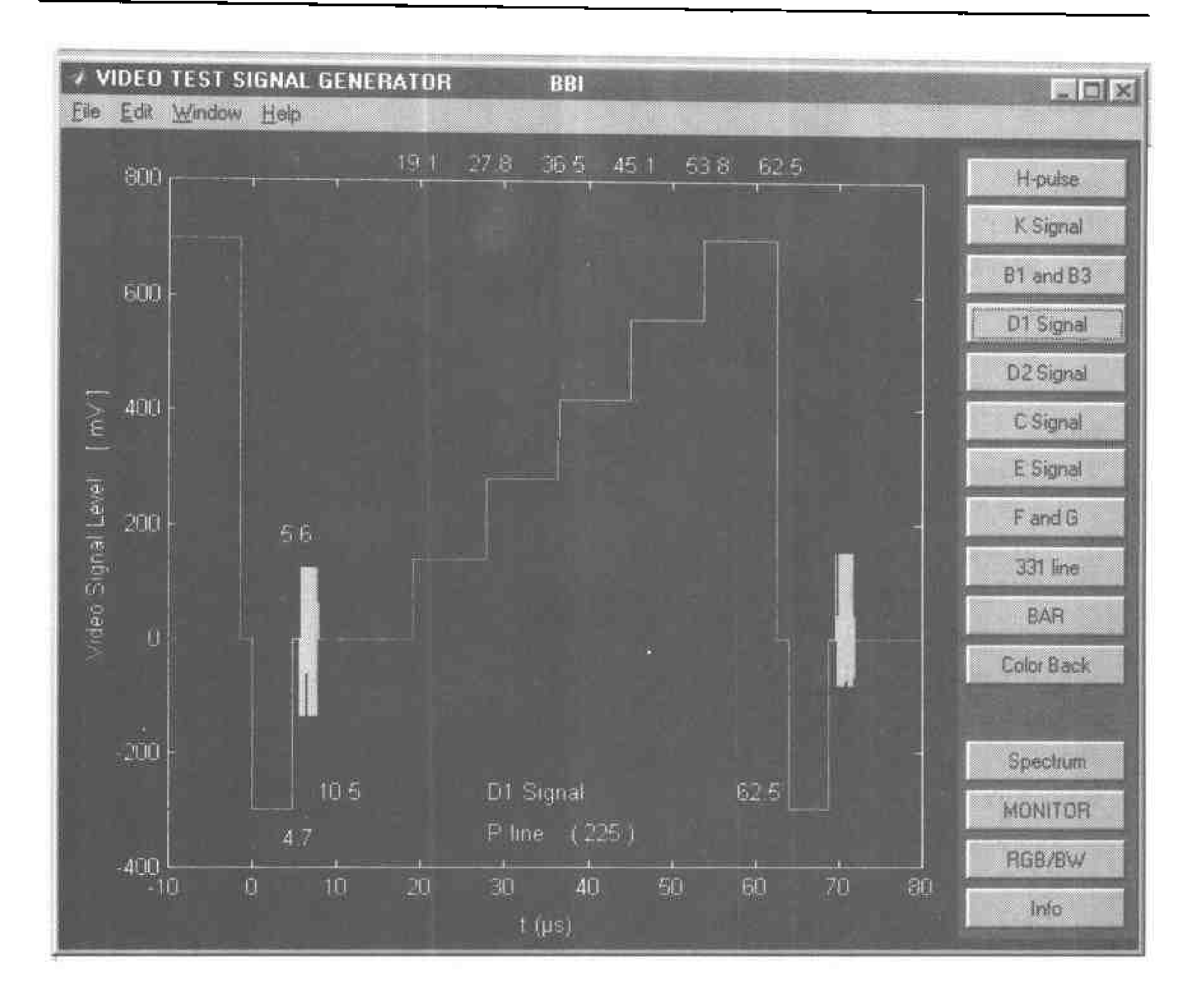


图 6.10　D1 信号的时域波形（图中的数字表示时间）

度信号之间的关系，还可以灵活地在彩色信号矢量和色度三角形之间进行切换，这样就将彩色电视中出现的色度信号与色度学中的色度坐标联系在一起了。使用仿真的彩色矢量示波器可以加深对彩色电视所涉及的色度学问题的理解。

颜色具有三个基本特性：色调、色饱和度和亮度，因此要表示一种颜色要使用三个参数，在电视机的显示电路中使用 R、G、B 三基色信号，在彩色电视信号的传输过程中使用 Y、U 和 V 信号（亮度信号和色差信号），另外计算机中通常使用 H、S 和 V，在印刷过程中使用 C、M、Y 和 K，而在色度学中则使用 X、Y 和 Z。

Y、U、V 与 R、G、B 之间的关系可以用以下的矩阵方式来表示，

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

在 PAL 制中，色度信号的向量表达式为，

$$C = \begin{cases} U + jV & (N) \\ U - jV & (P) \end{cases}$$

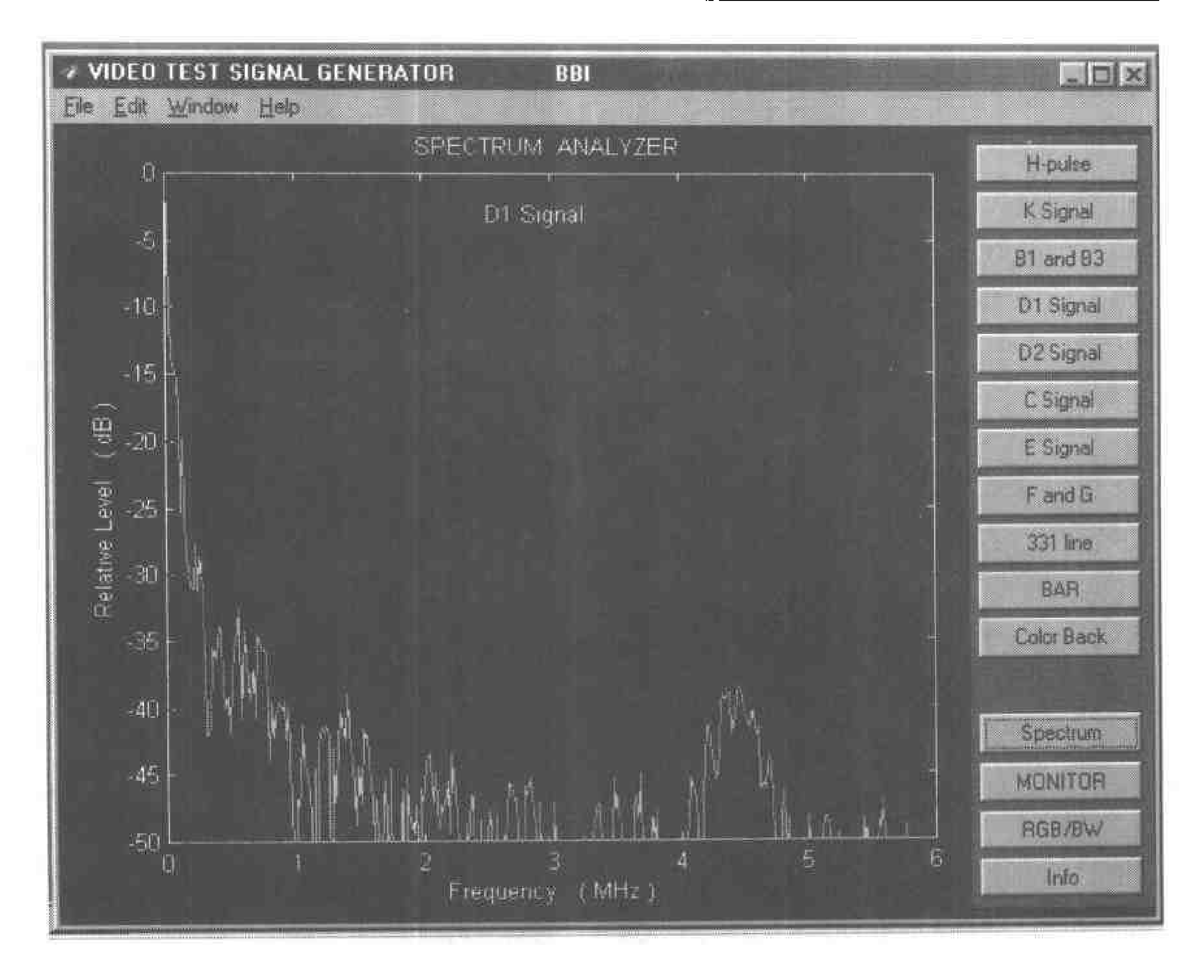


图 6.11 D1 信号的频谱

其中，字母 N 表示 N 行，字母 B 表示 B 行。使用 MATLAB 可以很容易表示出色度信号 C 与显象三基色信号 RBG 之间的关系，故而完成对彩色矢量示波器的仿真。

电视 RGB 制与 XYZ 计色制的关系为，

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.606 & 0.173 & 0.200 \\ 0.298 & 0.587 & 0.114 \\ 0 & 0.066 & 1.115 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

而颜色的色坐标 x 和 y 为，

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}$$

自然界中所有的颜色在直角坐标系 xy 内均位于一个舌形区域之内，这区域通常称为色度三角形。显然，使用 MATLAB 是很容易在彩色电视的色度信号与色度三角形之间进行转换的。

在各种版本的 MATLAB 中均设有现成的 M 函数 hsv2rgb.m 和 rgb2hsv.m，使用它们可以在 RGB（三基色）和 HSV（色调、饱和度、强度）之间进行转换。

自定义的 M 函数 vectorsc.m 用来对彩色矢量示波器进行仿真，它在 4.2 版和 5.x 版内

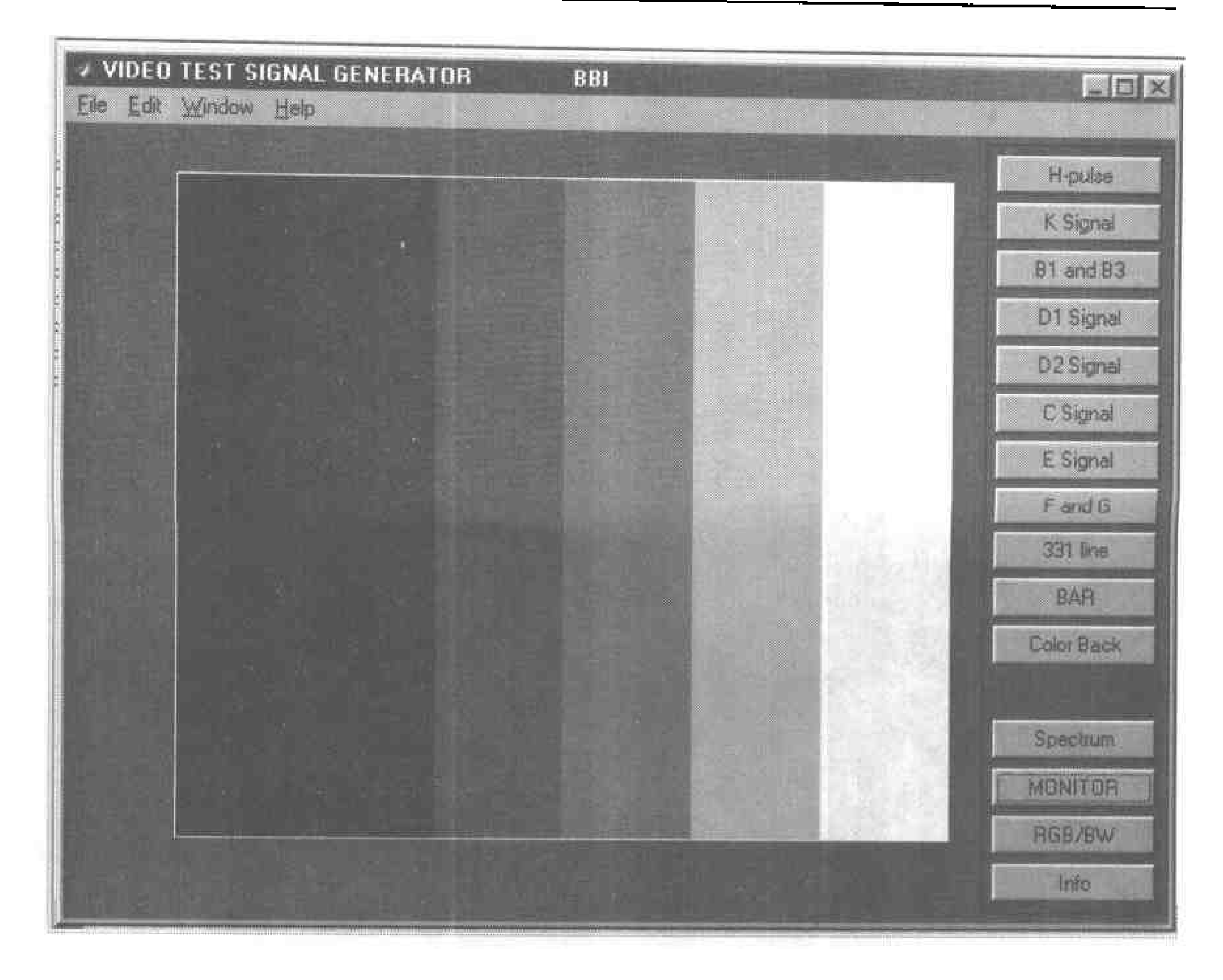


图 6.12 D1 信号所显示的图像

均可以运行。为了便于使用，在函数中采用了用户控制框来输入参数和进行控制。在仿真过程中，使用下拉式菜单可以对颜色、红、绿、蓝、饱和度、强度、色度信号的幅角等参数分别进行选择；也可使用编辑框对各种参数的数值进行赋值；另外还可以使用滑标对彩色、对比度和色调进行连续的调整；使用 Graph/Vector 按钮便可在彩色矢量示波器和色度图之间进行转换；而按下 Chrom Vector 按钮则可通过鼠标在矢量示波器或在色度三角形上直接对颜色进行选择。在函数的运行过程中，表示颜色的各种参数均在屏幕上显示出来，如 RGB、HSV、YUV、xy 等。色度图中舌形曲线边上出现的数字表示的是各个谱色的波长，单位为 nm（纳米）。

vectorsc.m 函数的程序清单如下：

```
function [rgbs] = vectorsc (action);
%
%       BBI       97/4
if nargin<1; action='initialize'; end;
global rgb rgbm H Hc actionLast kfine kcont kyc...
```

```
        rgba rgbs rgbs1 kgraph;
load color;
if strcmp (action,'initialize'),
    figNumber=figure ('Name','CHROMINANCE VECTORSCOPE BBI', ...
            'Num','off','pos', [5 29 792 530]);
    axes ('Units','normalized','Position', [0.1 0.1 0.7 0.84]);
    axis off; whitebg ('w');
    % ========================================
    % Information for all buttons
    labelColor= [0.8 0.8 0.8];
    top=0.985; left=0.84; bottom=0.025;
    btnWid=0.15; btnHt=0.05; spacing=0.007;
    % ========================================
    % The CONSOLE frame
    frmBorder=0.01; yPos=0.02-frmBorder;
    frmPos= [left-frmBorder yPos btnWid+2*frmBorder 0.94+4*frmBorder];
    cslHndl=uicontrol ('Style','frame','Units','normalized', ...
        'Pos', frmPos,'Back', [.2 .4 .4]);
    % ========================================
    % The COLOR button
    btnNumber=1;
    yPos=top- (btnNumber-1) * (btnHt+spacing);
    callbackStr='vectorsc (''Color'');';

    btnPos= [left yPos-btnHt btnWid btnHt];
    Hcolor=uicontrol ('Style','popup','Units','normalized', ...
        'Pos', btnPos,'String', P_str, ...
        'Back', [.98 .98 .98],'Callback', callbackStr);
    % ========================================
    % The Red button
    btnNumber=2;
    yPos=top- (btnNumber-1) * (btnHt+spacing);
    callbackStr='vectorsc (''Red'');';
    P_str=str2mat ('0','1','.95','.9','.85','.8','.75','.7');
    P_str=str2mat (P_str,'.65','.6','.55','.5','.45','.4','.35');
    P_str=str2mat (P_str,'.3','.25''.2','.15','.1','.05');

    btnPos= [left+.075 yPos-btnHt btnWid-.075 btnHt];
    uicontrol ('Style','text','Units','normalized', ...
```

```
        'Position', [left yPos-btnHt btnWid-.08 .046], ...
        'String','Red ','Back', [.8 .8 .8]);
uicontrol ('Style','popup','Units','normalized', ...
    'Pos', btnPos,'String', P_str, ...
    'Back', [.98 .98 .98],'Callback', callbackStr);
% ==========================================
% The Green button
btnNumber=3;
yPos=top- (btnNumber-1) * (btnHt+spacing);
callbackStr='vectorsc (''Green'');';
btnPos= [left+.075 yPos-btnHt btnWid-.075 btnHt];
uicontrol ('Style','text','Units','normalized', ...
        'Position', [left yPos-btnHt btnWid-.08 .046], ...
        'String','Green ','Back', [.8 .8 .8]);
uicontrol ('Style','popup','Units','normalized', ...
    'Pos', btnPos,'String', P_str, ...
    'Back', [.98 .98 .98],'Callback', callbackStr);
% ==========================================
% The Blue button
btnNumber=4;
yPos=top- (btnNumber-1) * (btnHt+spacing);
callbackStr='vectorsc (''Blue'');';
btnPos= [left+.075 yPos-btnHt btnWid-.075 btnHt];
uicontrol ('Style','text','Units','normalized', ...
        'Position', [left yPos-btnHt btnWid-.08 .046], ...
        'String','Blue ','Back', [.8 .8 .8]);
uicontrol ('Style','popup','Units','normalized', ...
        'Position', btnPos,'String', P_str, ...
        'Back', [.98 .98 .98],'Callback', callbackStr);
% ==========================================
% The Saturation button
btnNumber=5;
yPos=top- (btnNumber-1) * (btnHt+spacing);
callbackStr='vectorsc (''Saturation'');';
P_str=str2mat ('1','.95','.9','.85','.8','.75','.7');
P_str=str2mat (P_str,'.65','.6','.55','.5','.45','.4','.35');
P_str=str2mat (P_str, '.3','.25''.2','.15','.1','.05');

btnPos= [left+.075 yPos-btnHt btnWid-.075 btnHt];
```

```
uicontrol ('Style','text','Units','normalized', ...
    'Position', [left yPos-btnHt btnWid-.08 .046], ...
    'String','Saturat','BackgroundColor', [.8 .8 .8]);
uicontrol ('Style','popup','Units','normalized', ...
    'Position', btnPos,'String', P_str, ...
    'Back', [.98 .98 .98],'Callback', callbackStr);
% ==================================================
% The Value button
btnNumber=6;
yPos=top- (btnNumber-1) * (btnHt+spacing);
callbackStr='vectorsc (''Value'');';

btnPos= [left+.075 yPos-btnHt btnWid-.075 btnHt];
uicontrol ('Style','text','Units','normalized', ...
    'Position', [left yPos-btnHt btnWid-.08 .046], ...
    'String','Value ','BackgroundColor', [.8 .8 .8]);
uicontrol ('Style','popup','Units','normalized', ...
    'Position', btnPos,'String', P_str, ...
    'Back', [.98 .98 .98],'Callback', callbackStr);
% ==================================================
% The ANGLE button
btnNumber=7;
yPos=top- (btnNumber-1) * (btnHt+spacing);
callbackStr='vectorsc (''Angle'');';

btnPos= [left+.075 yPos-btnHt btnWid-.075 .046];
uicontrol ('Style','text','Units','normalized', ...
    'Position', [left yPos-btnHt btnWid-.08 .046], ...
    'String','Angle ','Back', [.8 .8 .8]);
uicontrol ('Style','edit','Units','normalized', ...
    'Position', btnPos,'String', num2str (103.5), ...
    'Back', [.98 .98 .98],'Callback', callbackStr);
% ==================================================
% The FINE button
btnNumber=8;
yPos=top- (btnNumber-1) * (btnHt+spacing);
callbackStr='vectorsc (''Fine'');';
P_str=str2mat ('R','G','B','Y','|C|','arg (C)','H');
P_str=str2mat (P_str,'S','V','x','y');
```

```
uicontrol ('Style','popup','Units','normalized', ...
    'Position', [left yPos-btnHt btnWid-.08 .046], ...
    'String', P_str,'Back', [.98 .98 .98], ...
    'Callback', callbackStr);
callbackStr='vectorsc (''Fine Value'');';
btnPos= [left+.075 yPos-btnHt btnWid-.075 .046];
uicontrol ('Style','edit','Units','normalized', ...
    'Position', btnPos,'String', num2str (.75), ...
    'Back', [.98 .98 .98],'Callback', callbackStr);

% ====================================
% The VECTOR hutton
btnNumber=9;
yPos=top- (btnNumber-1) * (btnHt+spacing);
btnPos= [left yPos-btnHt*1.05 btnWid btnHt];
uicontrol ('Style','push', Units','normalized', ...
    'Position', btnPos,'String','Chrom Vector', ...
    'Callback','vectorsc (''Vector'');');
% ====================================
% The Y/C/H button
btnNumber=10;
callbackStr='vectorsc (''Y/C/H'');';
P_str=str2mat ('Y: Contrast','|C|: Colour','Hue Adjust');
yPos=top- (btnNumber-1) * (btnHt+spacing);
uicontrol ('Style','popup','Units','normalized', ...
    'Position', [left yPos-btnHt btnWid .046], ...
    'String', P_str,'Back', [.98 .98 .98], ...
    'Callback', callbackStr);
% ====================================
% The ADJUST button
btnNumber=11;
yPos=top- (btnNumber-1) * (btnHt+spacing);
btnPos= [left yPos-btnHt*0.8 btnWid btnHt*.7];
uicontrol ('Units','normal','Position', btnPos, ...
        'Style','slider','min', 0,'max', 1, ...
        'callback','vectorsc (''Adjust'');','value', .618);
% ====================================
% The Store button
```

```
btnNumber = 12;
yPos = top - (btnNumber - 1) * (btnHt + spacing);
btnPos = [left yPos - btnHt * 1.05 btnWid btnHt];
uicontrol ('Style','push','Units','normalized', ...
    'Position', btnPos,'String','Store', ...
    'Callback',' [rgbs] = vectorsc (''Store'');');
% ====================================
% The COMPARE button
btnNumber = 13;
yPos = top - (btnNumber - 1) * (btnHt + spacing);
btnPos = [left yPos - btnHt * 1.05 btnWid btnHt];
uicontrol ('Style','push','Units','normalized', ...
    'Position', btnPos,'String','Compare', ...
    'Callback','vectorsc (''Compare'');');
% ====================================
% The RESTORE button
btnNumber = 14;
yPos = top - (btnNumber - 1) * (btnHt + spacing);
btnPos = [left yPos - btnHt * 1.05 htnWid btnHt];
uicontrol ('Style','push','Units','normalized', ...
    'Position', btnPos,'String','Restore', ...
    'Callback','vectorsc (''Restore'');');
% ====================================
% The C - GRAPH button
btnNumber = 15;
yPos = top - (btnNumber - 1) * (btnHt + spacing);
btnPos = [left yPos - btnHt * 1.05 btnWid btnHt];
uicontrol ('Style','push','Units','normalized', ...
    'Position', btnPos,'String','Graph/Vector', ...
    'Callback','vectorsc (''C - Graph'');');
% ====================================
% The INFO button
uicontrol ('Style','push','Units','normalized', ...
    'Position', [left bottom btnWid btnHt], ...
    'String','Info','Callback','vectorsc (''info'');');

% ====================================
kfine = 1; kcont = 0; kyc = 1; rgbs = []; kgraph = 1;
rgb = c_rgb (1, :); rgbm = rgb;
```

```
        t=vector1 (rgb, 1); clear t;

    elseif strcmp (action,'Color');
        rgbm=rgb; k=get (gco,'Value'); rgb=c_rgb (k,:);
        if kgraph>0; rgb=vector1 (rgb);
        elseif kgraph<0; rgb=cgraph (rgh);
        end;
    elseif strcmp (action,'Red');
        rgbm=rgb; k=get (gco,'Value');
        if k==1; rgb (1) =0;
        else; rgb (1) =1- (k-2) *.05;
        end;
        t=vector1 (rgb); clear t;
    elseif strcmp (action,'Green');
        rgbm=rgb; k=get (gco,'Value');
        if k==1; rgb (2) =0;
        else; rgb (2) =1- (k-2) *.05;
        end;
        t=vector1 (rgb); clear t;
    elseif strcmp (action,'Blue');
        rgbm=rgb; k=get (gco,'Value');
        if k==1; rgb (3) =0;
        else; rgb (3) =1- (k-2) *.05;
        end;
        t=vector1 (rgb); clear t;
    elseif strcmp (action,'Saturation');
        rgbm=rgb; k=get (gco,'Value'); hsv=rgb2hsv (rgb);
        hsv (2) =1- (k-1) *.05; rgb=hsv2rgb (hsv);
        if kgraph>0; rgb=vector1 (rgb);
        elseif kgraph<0; rgb=cgraph (rgb);
        end;
    elseif strcmp (action,'Value');
        rgbm=rgb; k=get (gco,'Value');
        hsv=rgb2hsv (rgh); hsv (3) =1- (k-1) *.05; rgb=hsv2rgb (hsv);
        if kgraph>0; rgb=vector1 (rgb);
        elseif kgrapb<0; rgb=cgraph (rgb);
        end;
    elseif strcmp (action,'Angle');
        rgbm=rgb; phi=get (gco,'String');
```

```
        phi=str2num (phi); rgb=hue2rgb (phi);
        t=vector1 (rgb); clear t;
elseif strcmp (action,'Vector');
        rgbm=rgb;
        if kgraph>0; rgb=vector1 (rgb, 0, 1);
        elseif kgraph<0; rgb=cgraph (rgb, 1);
        end;
elseif strcmp (action,'Fine');
        kfine=get (gco,'Value');
elseif strcmp (action,'Fine Value');
        rgbm=rgb; rgbf=rgb;
        value=get (gco,'String'); value=str2num (value);
        if kfine==1;            [WB] rgbf (1) =value;
        elseif kfine==2; [DW] rgbf (2) =value;
        elseif kfine==3; [DW] rgbf (3) =value;
        elseif kfine==4;
            yuv=rgb2yuv (rgbf); yuv (1) =value; rgbf=yuv2rgb (yuv);
        elseif kfine==5;
            yuv=rgb2yuv (rgbf); c=yuv (2) +yuv (3) *1i;
            cm=value; arg=angle (c);
            yuv (2) =cm*cos (arg); yuv (3) =cm*sin (arg); rgbf=yuv2rgb (yuv);
        elseif kfine==6;
            yuv=rgb2yuv (rgbf); c=yuv (2) +yuv (3) *1i;
            cm=abs (c); arg=value*pi/180;
            yuv (2) =cm*cos (arg); yuv (3) =cm*sin (arg); rgbf=yuv2rgb (yuv);
        elseif kfine==7;
            hsvf=rgb2hsv (rgbf); hsvf (1) =value; rgbf=hsv2rgb (hsvf);
        elseif kfine==8;
            hsvf=rgb2hsv (rgbf); hsvf (2) =value; rgbf=hsv2rgh (hsvf);
        elseif kfine==9;
            hsvf=rgh2hsv (rgbf); hsvf (3) =value; rgbf=hsv2rgb (hsvf);
        elseif kfine==10;
            xy=rgb2xy (rgbf); xy (1) =value; rgbf=xy2rgb (xy);
        elseif kfine==11;
            xy=rgb2xy (rgbf); xy (2) =value; rgbf=xy2rgb (xy);
        end;
        I=find (rgbf>1); J=find (rgbf<0);
        if length (I) ==0 & length (J) ==0;
            if kgraph>0; rgb=vector1 (rgbf);
```

```
            elseif kgraph<0; rgb=cgraph (rgbf);
            end;
        end;
    elseif strcmp (action,'Y/C/H');
        kyc=get (gco,'Value');
    elseif strcmp (action,'Adjust');
        k=get (gco,'Value');
        if kyc= =1;
            del=1-max (rgb); ymax=rgb+del;
            del=sort (rgb); del=del (1); ymin=rgb;
            if del>1e-3; ymin=rgb-del; end;
            rgba= (ymax-ymin) *k+ymin; t=vector1 ( [rgba; rgb]); clear t;
        elseif kyc= =2;
            yuv=rgb2yuv (rgb); cmin=yuv (2: 3) *.05; cmax=yuv (2: 3) *20;
            for i=105: 5: 2000;
                rgbc=yuv2rgb ( [yuv (1) yuv (2: 3) *.01*i]);
                if max (rgbc) >1 | min (rgbc) <1e-5;
                    cmax=yuv (2: 3) *.01* (i-5); break;
                end;
            end;
            for i=99: -1: 5;
                rgbc=yuv2rgb ( [yuv (1) yuv (2: 3) *.01*i]);
                if max (rgbc) >1 | min (rgbc) <1e-5;
                    cmin=yuv (2: 3) *.01* (i+1); break;
                end;
            end;
            ck= (cmax-cmin) *k+cmin; rgba=yuv2rgb ( [yuv (1), ck]);
            t=vector1 ( [rgba; rgb]); clear t;
        elseif kyc= =3;
            h=rgb2hsv (rgb); h (1) =k; rgba=hsv2rgb (h);
            t=vector1 ( [rgba; rgb]);
        end; rgb=rgba;
    elseif strcmp (action,'Store');
        rgbs= [rgbs; rgbs1]; get (gco);
    elseif strcmp (action,'H+');
        rgbm=rgb; del=1/60;
        h=rgb2hsv (rgb); h (1) =h (1) +del; h (1) =rem (h (1), 1);
        rgb=hsv2rgb (h); clear t;
        if kgraph>0; rgb=vector1 (rgb);
```

```
        elseif kgraph<0; rgb=cgraph (rgb);
        end;
    elseif strcmp (action,'H-');
        rgbm=rgb; del=1/60;
        h=rgb2hsv (rgh); h (1) =h (1) -del; h (1) =rem (h (1) +1, 1);
        rgb=hsv2rgb (h);
        if kgraph>0; rgb=vector1 (rgb);
        elseif kgraph<0; rgb=cgraph (rgb);
        end;
    elseif strcmp (action,'Compare');
        if strcmp (actionLast,'Adjust'); cmap= [rgba; rgbm];
        else; cmap= [rgb; rghm];
        end;
        if strcmp (actionLast,'Compare') = =0;
            Hc=suhplot ('position', [.65 .20 .16 .15]);
            image ( [1 2]), colormap (cmap);
            hold on; v=axis; v1=mean (v (1: 2));
            plot ( [v1 v1], [v (3) v (4)],'k'); hold off;
            set (gca,'XTick', [],'YTick', []);
        end;
    elseif strcmp (action,'Restore');
        if kgraph>0; rgh=vector1 (rgb);
        elseif kgraph<0; rgb=cgraph (rgb);
        end;
    elseif strcmp (action,'C-Graph');
        if kgraph<0; t=vector1 (rgb); clear t;
        elseif kgraph>0; cgraph (rgb);
        end; kgraph= -1 * kgraph;

    elseif strcmp (action,'info'),
        ttlStr=get (gcf,'Name');
        hlpStr= ...
            ['This window simulates a chrominance vectorscope.                  '
             'The chrominance sigal C is consists of U and V.                   '
             ' In PAL system:                                                   '
             '                                                                  '
             ' Y = 0.2990 * R + 0.5870 * G + 0.1140 * B                         '
             ' U = -0.1474 * R - 0.2894 * G + 0.4368 * B                        '
             ' V = 0.6148 * R - 0.5148 * G - 0.1000 * B                         '
```

```
        '                                                      '
        ' [I] 123 [Q] 33 - [I] 303 - [Q] 213                   '
        '                                                      '
        ' [G-Y] 236.5 - [G-Y] 56.5 [G-Y] =0 146.5, 326.5       '
        '                                                      '
        'The saturation and value of a color may vary in a     '
        'specific range when the C signal is fixed, as the Y   '
        'signal can be changed.                                '
        '                                                      '
        'The values of x and y indicate the saturation and     '
        'the hue characteristics in a specific color, but      '
        'they have no relation with the value, which only      '
        'presents the strength , in the HSV system.            '
        '                                                      '
        'Output variable name: rgbs                            '
        '                                                      '
        'File name: vectorsc.m , vector1.m , hue2rgb.m         '
        ' rgb2yuv.m , yuv2rgb.m , rgb2xy.m & xy2rgb.m          '];
    helpfun (ttlStr, hlpStr);
end;
if strcmp (actionLast,'Compare');
    if exist ('Hc'); delete (Hc); end;
    if strcmp (action,'Compare'); action='Compare_II'; end;
end;
actionLast=action;       kcont=kcont+1;
```

自定义的 M 函数 vectorsc.m 在运行的过程中要调用以下的 M 函数和 MAT 数据文件：

hue2rgb.m	rgb2yuv.m	xy2rgb.m	cgraph.m	linedisp.m
vector1.m	yuv2rgb.m	rgb2xy.m	COLOR.MAT	CIE.MAT

M 函数 hue2rgb.m 的程序清单如下：

```
function [rgb] =hue2rgh (alfa, am)
%
%      alfa --- Angle of the Chrominance Signal C = U + j V
%       am --- Amplitute of the Chrominance Signal C. And the default
%       value of am is calculeted in the condition of 100 % saturation.
%
%      BBI          97/04
```

```
if nargin<1;          alfa=103.5; end;
if alfa>=360;         alfa=alfa-360;
elseif alfa<0;        alfa=alfa+360;
end;
t=[0.299 0.587 0.114; -0.1474 -0.2894 0.4368; 0.6148 -0.5148 -0.1];
uv=[0.2894 0.5148; -0.1474 0.6148; -0.4368 0.1; -0.2894 -0.5148];
uv=[uv; 0.1474 -0.6148; 0.4368 -0.1];
if nargin<2;
    if alfa>=347.1077 | alfa<60.6578;
        x1=uv(1, 1); y1=uv(1, 2); x2=uv(6, 1); y2=uv(6, 2);
    elseif alfa>=60.6578 & alfa<103.4834;
        x1=uv(2, 1); y1=uv(2, 2); x2=uv(1, 1); y2=uv(1, 2);
    elseif alfa>=103.4834 & alfa<167.1077;
        x1=uv(3, 1); y1=uv(3, 2); x2=uv(2, 1); y2=uv(2, 2);
    elseif alfa>=167.1077 & alfa<240.6578;
        x1=uv(4, 1); y1=uv(4, 2); x2=uv(3, 1); y2=uv(3, 2);
    elseif alfa>=240.6578 & alfa<283.4834;
        x1=uv(5, 1); y1=uv(5, 2); x2=uv(4, 1); y2=uv(4, 2);
    elseif alfa>=283.4834 & alfa<347.1077;
        x1=uv(6, 1); y1=uv(6, 2); x2=uv(5, 1); y2=uv(5, 2);
    end;
    k=(y2-y1)/(x2-x1); am=y1-k*x1; alf=alfa*pi/180;
    sina=sin(alf); cosa=cos(alf); am=am/(sina-k*cos(alf));
    u=am*cosa; v=am*sina; c=[u v]';
else;
    alf=alfa*pi/180; sina=sin(alf); cosa=cos(alf);
    u=am*cosa; v=am*sina; c=[u v]';
end;
if alfa<=103.4834 | alfa>347.1077;
    g=0; t=[t(2, 1) t(2, 3); t(3, 1) t(3, 3)]; tt=inv(t);
    r=tt(1,:)*c; b=tt(2,:)*c;
elseif alfa<=240.6578 & alfa>103.4834;
    b=0; t=[t(2, 1:2); t(3, 1:2)]; tt=inv(t);
    r=tt(1,:)*c; g=tt(2,:)*c;
elseif alfa<=347.1077 & alfa>240.6578;
    r=0; t=[t(2, 2:3); t(3, 2:3)]; tt=inv(t);
    g=tt(1,:)*c; b=tt(2,:)*c;
end;
I=find(r>1); r(I)=ones(size(I)); I=find(r<0); r(I)=zeros(size(I));
```

```
I=find (g>1); g (I) =ones (size (I)); I=find (g<0); g (I) =zeros (size (I));
I=find (b>1); b (I) =ones (size (I)); I=find (b<0); b (I) =zeros (size (I));
rgb= [r g b];
```

M函数 rgb2yuv.m 的程序清单如下：

```
function [yuv, c] =rgb2yuv (rgb);
%
% BBI 97/04/12
r=rgb (:, 1); g=rgb (:, 2); b=rgb (:, 3);
y=.299*r+.587*g+.114*b; u=.493* (b-y); v=.877* (r-y);
a=u+v*1i; argc=angle (a) *180/pi;
if argc<0; argc=argc+360; end;
yuv= [y u v]; c= [abs (a) argc];
```

M函数 yuv2rgb.m 的程序清单如下：

```
function [rgb, sat] =yuv2rgb (yuv);
%
%    BBI    97/04/12

y=yuv (:, 1); u=yuv (:, 2); v=yuv (:, 3);
r=v/.877+y; b=u/.493+y; g= (y-.299*r-.114*b) /.587;
maxr=max (r); maxg=max (g); maxb=max (b);
rgb= [r g b]; sat=max ( [maxr maxg maxb]);
```

M函数 rgb2xy.m 的程序清单如下：

```
function [xy] =rgb2xy (rgb)
%
% BBI 97/04
T= [.606 .173 .2; .299 .587 .114; 0 .066 1.115];
xyz=rgb*T'; xyz=xyz/sum (xyz);
xy=xyz (1: 2); xy=round (xy*1000) /1000;
```

M函数 xy2rgb.m 的程序清单如下：

```
function [rgb] =xy2rgb (xy)
%
```

```
% BBI 97/04
T= [.606 .173 .2; .299 .587 .114; 0 .066 1.115];
T=inv (T); xyz= [xy 1-sum (xy)];
rgb=xyz * T'; rgbm=max (rgb); rgb=rgb/rgbm;
```

M函数vector1.m的程序清单如下:

```
function [rgb] =vector1 (rgb, kc, kv)
%
% BBI 97/04
global H H1 rgbm rgbs1;
if nargin<1; rgb= [1 0 0]; end;
if nargin<3; kv=0; end;
if nargin<2; kc=0; end;
c=rgb2yuv (rgb (1,:)); rgbs1=rgb (1,:);
a=c (2) +c (3) * 1i; ph=angle (a) * 180/pi;
if pb<0; ph=360+ph; end;
a=abs (a); a=round (a*1000) /1000; c=round (c*1000) /1000;
r=rgb (1, 1); r=round (r*1000) /1000;
g=rgb (1, 2); g=round (g*1000) /1000;
b=rgb (1, 3); b=round (b*1000) /1000;
h=rgb2hsv (rgb (1,:)); h=round (h*1000) /1000;
xy=rgb2xy (rgb (1,:));
astr=num2str (a); phstr=num2str (ph);
c1str=num2str (c (1)); c2str=num2str (c (2)); c3str=num2str (c (3));
rstr=num2str (r); gstr=num2str (g); bstr=num2str (b);
h1str=num2str (b (1)); h2str=num2str (h (2)); h3str=num2str (h (3));
xstr=num2str (xy (1)); ystr=num2str (xy (2));
if kc= =0; delete (H); end;
H=subplot ('position', [.1 .15 .5 .7]);
a=1: 91; a= (a-1) *4*pi/180;
x=sin (a); y=cos (a);
plot (x, y,'k');
axis equal; axis off; hold on;
plot ( [-1.0 1.0], [0 0],'k:');
plot ( [0 0], [-1.0 1.0],'k:');
a=1: 12; a= (a-1) *pi/6;
x1=.97*sin (a); x2=1.03*sin (a);
y1=.97*cos (a); y2=1.03*cos (a);
```

```
for i=1: 12;
     plot ( [x1 (i) x2 (i)], [y1 (i) y2 (i)],'k');
end;
tri= [1 0 0; 1 1 0; 0 1 0; 0 1 1; 0 0 1; 1 0 1; 1 0 0];
tri=rgb2yuv (tri) *1.5;
plot (tri (:, 2), tri (:, 3),'b:');
for i=1: 6;
     plot (tri (i, 2), tri (i, 3),'b+');
end;
ph1=55.8548; gy1=hue2rgb (ph1+90); gy1=rgb2yuv (gy1) *1.5;
gy2=hue2rgb (ph1+270); gy2=rgb2yuv (gy2) *1.5;
plot ( [gy1 (2) gy2 (2)], [gy1 (3) gy2 (3)],'b:');

GY_1=hue2rgb (ph1); GY_1=rgb2yuv (GY_1) *1.5;
GY_2=hue2rgb (ph1+180); GY_2=rgb2yuv (GY_2) *1.5;
plot ( [GY_1 (2) GY_2 (2)], [GY_1 (3) GY_2 (3)],'b:');
q1=hue2rgb (33); q1=rgb2yuv (q1) *1.5;
q2=hue2rgb (213); q2=rgb2yuv (q2) *1.5;
plot ( [q1 (2) q2 (2)], [q1 (3) q2 (3)],'b:');
i1=hue2rgb (123); i1=rgb2yuv (i1) *1.5;
i2=hue2rgh (303); i2=rgb2yuv (i2) *1.5;
plot ( [i1 (2) i2 (2)], [i1 (3) i2 (3)],'b:'); plot (0, 0,'k+');
text (1.15, 0,'U'); text (-.01, 1.25,'V');
text (1.05, 0,'0'); text (-1.2, 0,'180');
text (-.03, 1.1,'90'); text (-.05, -1.1,'270');
text (.91, .55,'30'); text (.5, .95,'60');
text (-1.05, .56,'150'); text (-.6, .96,'120');
text (.91, -.55,'330'); text (.5, -.95,'300');
text (-1.05, -.56,'210'); text (-.6, -.96,'240');
text (-.38, .86,'R'); text (-.46, -.84,'G'); text (.7, -.16,'B');
text (-.67, -.68,'G-Y'); text (.58, .35,'Q'); text (-.45, .66,'I');
text (-.84, .39,'G-Y=0');
plot ( [0 c (2) *1.5], [0, c (3) *1.5],'r');
text (-1.3, 1.4, ['C = [ ' astr ' ' phstr ' (deg) ]']);
text (-1.3, 1.2, ['YUV = [ ' c1str ' ' c2str ' ' c3str ' ]']);
text (-1.3, -1.3, ['RGB = [ ' rstr ' ' gstr ' ' bstr ' ]']);
text (0.5, -1.3, ['HSV = [ ' h1str ' ' h2str ' ' h3str ' ]']);
text (.5, 1.4, ['xy = [ ' xstr ' ' ystr ' ]']);
if kv==1; c=ginput (1); end;
```

```
hold off; set (gcf,'color','w');
% =====================================
    H1=subplot ('position', [.65 .75 .16 .15]);
    linedisp (rgb); hold on;
% =====================================
if kv==1;
    load pal;
    c=c/1.5; % cc=c (1) +c (2) *1i;
    tt=inv (t); tt=tt';
    for y=1: -.01: 0.1; % Y = Ymax
        rgbv= [y c] *tt;
        I=find (rgbv>1); J=find (rgbv<0);
        if length (I) ==0 & length (J) ==0;
            vector1 (rgbv); rgb=rgbv;
            break;
        end;
    end;

end;
```

M 函数 linedisp.m 的程序清单如下：

```
function [] =linedisp (x, map, astr)
%
% BBI 97/03/15
if nargin<3; astr=' '; end;
if nargin<2; map=x; I=size (map); x=1: I (1); end;
I=find (map<1e-6); map (I) =zeros (size (I));
image (x), colormap (map); set (gca,'XTick', [],'YTick', []);
```

M 函数 cgraph.m 的程序清单如下：

```
function [rgb] =cgraph (rgb, kxy, str)
%
% BBI 97/04
global H H1;
if nargin<3; str='NTSC'; end;
if nargin<2; kxy=0; end;
if nargin<1; rgb= [.2 .7 .4]; end;
```

```
load cie;
delete (H); H=subplot ('position', [.1 .1 .5 .8]);
xy= [xy; xy (1,:)]; wv= [450 470 480: 10: 620];
kwv= [10 14 16 18 20 22 26: 2: 46];
cxy=rgb2xy (rgb); cxy=round (cxy * 1000) /1000;
xys= [num2str (cxy (1)) ' ' num2str (cxy (2))];
hsvs=rgb2hsv (rgb); hsvs=round (hsvs * 1000) /1000;
hs= [num2str (hsvs (1)) ' ' num2str (hsvs (2)) ' ' num2str (hsvs (3))];
plot (xy (:, 1), xy (:, 2),'r'); grid; hold on;
set (gca,'Aspect', [1 1],'Xtick', [0 .1 .2 .3 .4 .5 .6 .7]);
title ( ['COLOUR GRAPH --- ' str ' [C]: [.310 .316]']);
xlabel ( ['x xy: [ ' xys ' ] HSV: [ ' hs ' ]']);
ylabel ('y');
if strcmp (str,'NTSC');
    R= [.67 .33]; G= [.21 .71]; B= [.14 .08]; W= [.31 .316];
elseif strcmp (str,'PAL');
    R= [.64 .33]; G= [.29 .60]; B= [.15 .06]; W= [.313 .329];
end;
plot ( [R (1) G (1)], [R (2) G (2)],'b');
plot ( [R (1) B (1)], [R (2) B (2)],'b');
plot ( [B (1) G (1)], [B (2) G (2)],'b');
nwv=length (kwv);
for i=1: nwv;
    ki=kwv (i); plot (xy (ki, 1), xy (ki, 2),'r+');
end;
for i=1: 3;
    ki=kwv (i);
    text (xy (ki, 1) -.075, xy (ki, 2), num2str (wv (i)),'color','r');
end;
ki=kwv (4); text (xy (ki, 1) -.075, xy (ki, 2) -.04, num2str (wv (4)),'col-
or','r');
ki=kwv (5); text (xy (ki, 1) -.075, xy (ki, 2), num2str (wv (5)),'color','r');
ki=kwv (6); text (xy (ki, 1) -.075, xy (ki, 2), num2str (wv (5)),'color','r');
text (xy (kwv (7), 1) -.1, .85, num2str (wv (7)),'color','r');
for i=8: nwv-1;
    ki=kwv (i);
    text (xy (ki, 1) +.04, xy (ki, 2), num2str (wv (i)),'color','r');
end;
plot (xy (1, 1), xy (1, 2),'r+', xy (59, 1), xy (59, 2),'r+');
```

```
text (.65, .2,'700780');        text (.25, .02,'380400');
text (.18, .03,'V');            text (.06, .3,'C');
text (.11, .14,'B');            text (.04, .78,'G');
text (.42, .14,'M');            text (.69, .27,'R');
text (.52, .52,'Y');            text (.59, .45,'O');
text (.81, .60,'Violet:');      text (.95, .60,'390 - 450 nm');
text (.81, .55,'Blue:');        text (.95, .55,'450 - 500 nm');
text (.81, .50,'Green:');       text (.95, .50,'500 - 570 nm');
text (.81, .45,'Yellow:');      text (.95, .45,'570 - 590 nm');
text (.81, .40,'Orange:');      text (.95, .40,'590 - 610 nm');
text (.81, .35,'Red:');         text (.95, .35,'610 - 770 nm');
rgbt=xy2rgb (cxy); ht=rgb2hsv (rgbt); ht (2) =1;
rgbt=hsv2rgb (ht); xyt=rgb2xy (rgbt);
plot ( [W (1) (xyt (1) -W (1)) *1.2+W (1)], [W (2) (xyt (2) -W (2)) *
1.2+W (2)],'b:');
plot (W (1), W (2),'m+'); plot (cxy (1), cxy (2),'r+');
plot (R (1), R (2),'bx', G (1), G (2),'bx', B (1), B (2),'bx');
C=rgb2xy ( [0 1 1]); plot (C (1), C (2),'bx');
Y=rgb2xy ( [1 1 0]); plot (Y (1), Y (2),'bx');
M=rgb2xy ( [1 0 1]); plot (M (1), M (2),'bx');
if kxy= =1;
    xy1=ginput (1); rgbc=xy2rgb (xy1);
    I=find (rgbc>1); J=find (rgbc<0);
    if length (I) = =0 & length (J) = =0;
        rgb=rgbc; cxy=round (xy1*1000) /1000;
        rgbt=xy2rgb (cxy); ht=rgb2hsv (rgbt); ht=round (ht*1000) /1000;
        hs= [num2str (ht (1)) ' ' num2str (ht (2)) ' ' num2str (ht (3))];
        ht (2) =1; rgbt=hsv2rgb (ht); xyt=rgb2xy (rgbt);
        plot ( [W (1) (xyt (1) -W (1)) *1.2+W (1)], ...
            [W (2) (xyt (2) -W (2)) *1.2+W (2)],'b:');
        plot (xy1 (1), xy1 (2),'k+'); plot (W (1), W (2),'r+');
        xys= [num2str (cxy (1)) ' ' num2str (cxy (2))];
        xlabel ( ['x xy: [ ' xys ' ] HSV: [ ' hs ' ]']);
        ylabel ('y');
        delete (H1); H1=subplot ('position', [.65 .75 .16 .15]);
        linedisp (rgb (1,:));
    else;
        delete (H1); H1=subplot ('position', [.65 .75 .16 .15]);
        linedisp ( [.1 .1 .1]);
```

```
      end;
  elseif kxy==0;
        delete (H1); H1=subplot ('position', [.65 .75 .16 .15]);
        linedisp (rgb (1,:));
  end;
```

数据文件 CIE.MAT 中储存的变量名为 xy，它是 59×2 的矩阵，用于绘制色度图中的舌形曲线，

```
1.7313e-001 4.4776e-003
1.7255e-001 4.7600e-003
1.7202e-001 4.8770e-003
1.7143e-001 5.1020e-003
1.7031e-001 5.7881e-003
1.6888e-001 6.9002e-003
1.6690e-001 8.5353e-003
1.6678e-001 1.1013e-002
1.6112e-001 1.3793e-002
1.5664e-001 1.7705e-002
1.5099e-001 2.2740e-002
1.4396e-001 2.9703e-002
1.3550e-001 3.9879e-002
1.2414e-001 5.7814e-002
1.0959e-001 8.6843e-002
9.1256e-002 1.3268e-001
6.8761e-002 2.0071e-001
4.5377e-002 2.9495e-001
2.3460e-002 4.1270e-001
8.1680e-003 5.3842e-001
3.8585e-003 6.5482e-001
1.3870e-002 7.5019e-001
2.3834e-002 7.8539e-001
3.8852e-002 8.1202e-001
5.4922e-002 8.2882e-001
7.4331e-002 8.3384e-001
1.1415e-001 8.2616e-001
1.5472e-001 8.0583e-001
1.9284e-001 7.8170e-001
2.2962e-001 7.5433e-001
```

```
2.6578e-001 7.2432e-001
3.0158e-001 6.9237e-001
3.3735e-001 6.5889e-001
3.7310e-001 6.2445e-001
4.0875e-001 5.8962e-001
4.4406e-001 5.5471e-001
4.7957e-001 5.1941e-001
5.1247e-001 4.8658e-001
5.4479e-001 4.5443e-001
5.7515e-001 4.2423e-001
6.0293e-001 3.9650e-001
6.2704e-001 3.7249e-001
6.4823e-001 3.5139e-001
6.6578e-001 3.3402e-001
6.8010e-001 3.1976e-001
6.9149e-001 3.0835e-001
7.0061e-001 2.9930e-001
7.1906e-001 2.8094e-001
7.2305e-001 2.7695e-001
7.2599e-001 2.7401e-001
7.2827e-001 2.7173e-001
7.2997e-001 2.7003e-001
7.3100e-001 2.6900e-001
7.3199e-001 2.6801e-001
7.3272e-001 2.6728e-001
7.3354e-001 2.6646e-001
7.3438e-001 2.6563e-001
7.3463e-001 2.6537e-001
7.3488e-001 2.6512e-001
```

数据文件 COLOR.MAT 中储存的变量为 P_str 和 c_rgb，它们是配合 vectorsc.m 函数中的 color 按钮使用的，用于描述若干特定的颜色。

```
P_str =                          c_rgb =

        Amaranth                         0.5000      0        0.5000
        Aquamarine                       0.4482    0.9000     0.7482
        Azure                            0.2000      0        1.0000
        Blue                               0         0        0.5000
```

Brown	0.5600	0.3400	0.2300
Cyan	0	1.0000	1.0000
Dark Green	0.2210	0.3820	0.2210
Gray	0.7000	0.7000	0.7000
Green	0	1.0000	0
Magenta	1.0000	0	1.0000
Magenta CIE	1.0000	0	0.7740
Orange	1.0000	0.3277	0
Pink	1.0000	0.4000	0.6000
Purple	0.6100	0	0.9000
Red	1.0000	0	0
Yellow	1.0000	1.0000	0
Yellow CIE	1.0000	0.7410	0
Violet	0.2540	0	0.7000

图 6.13 和图 6.14 是 M 函数 vectorsc.m 的运行结果。

6.7 锁相环

由于锁相环路具有窄带、跟踪和稳定等性能，因此锁相技术在电子系统中得到了广泛的使用，如在卫星接收机和电视接收机就普遍地采用了锁相环（PLL）。锁相环由三个基本单元组成，它们是：鉴相器、低通滤波器（LPF）和压控振荡器（VCO），从工作原理上来说，锁相环本身是一个相位反馈系统或频率反馈系统。锁相环的模型如图 6.15 所示。

输入信号 $v_i(t)$ 和输出信号 $v_o(t)$ 同时输入鉴相器，在鉴相器中对两路信号的相位进行比较，然后生成相位误差电压 $v_e(t)$，鉴相器比较简单的模型就是乘法器。锁相环路内的低通滤波器用于滤除相位误差电压中的高频成分和噪声。压控振荡器的振荡频率受相位误差电压的控制，当输入的电压为零时，其振荡频率为压控振荡器的固有频率 f_0，当输入的电压为正电压时，压控振荡器的振荡频率高于其固有频率，而输入的电压为负电压时，压控振荡器的振荡频率则低于其固有频率；另外，振荡频率的变化是正比于误差电压的。在使用宽带滤波器的情况下，当压控振荡器的输出频率能够跟踪上输入频率的瞬时变化，就称为锁定状态；另外当使用窄带滤波器时，压控振荡器的输出频率能够跟踪上输入信号的平均频率时，也称为锁定状态。

为了对锁相环进行仿真，设输入信号为：$v_i(t)=\sin\omega_i t$，其中 ω_i 为输入信号的角频率，并且设输入信号的初相角为零；K_d 为鉴相器的增益常数；设压控振荡器的输出为 $v_0(t)=\cos[\omega_o t+\theta(t)]$，其中 ω_o 为压控振荡器固有的振荡角频率，而压控振荡器输出信

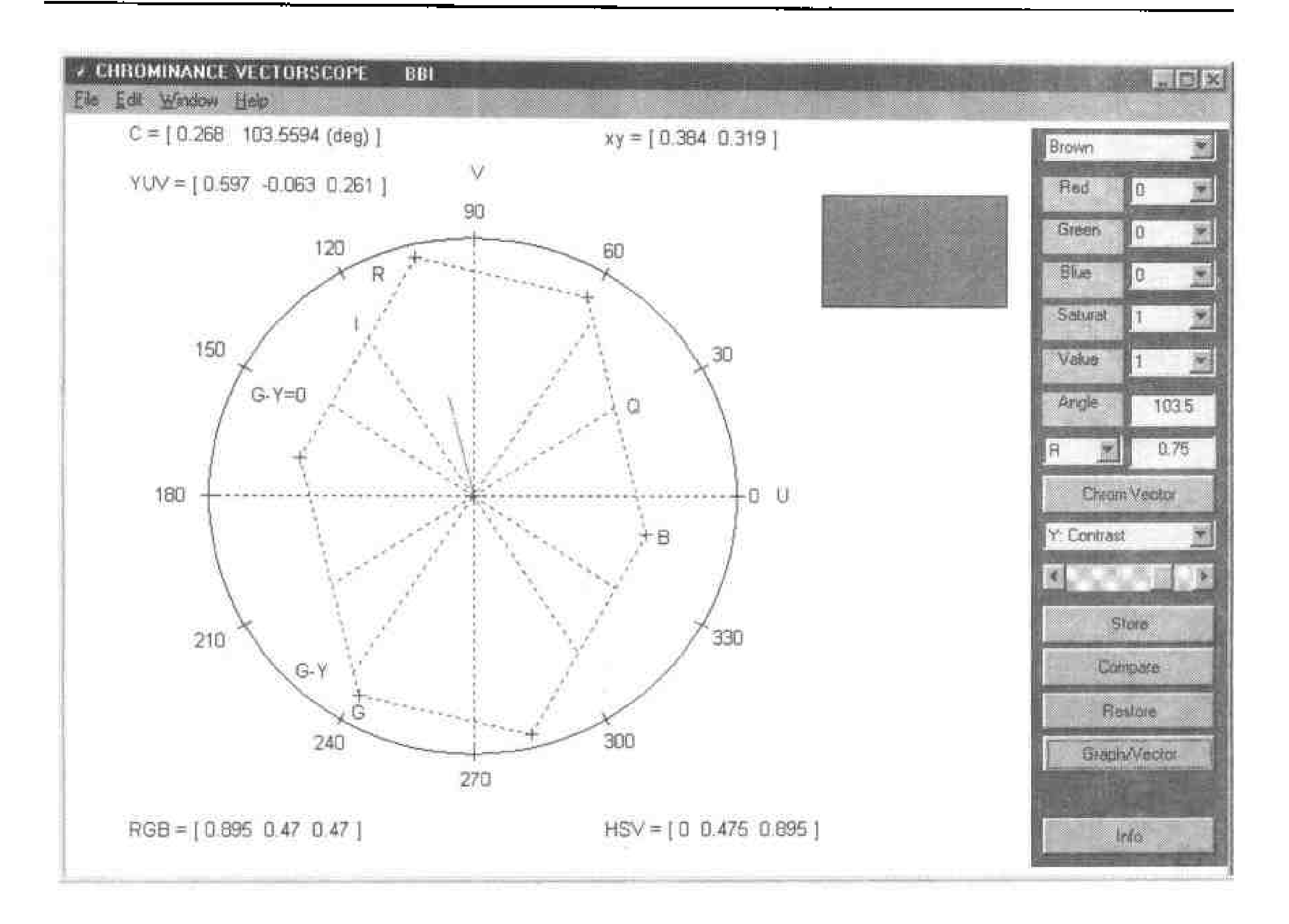


图 6.13　彩色矢量示波器

号的瞬时相位与其输入电压 v_c（t）之间的关系为：$\theta(t) = K_v \int_{-\infty}^{t} v_c(\tau)\, d\tau$，其中 K_v 称为 压控振荡器的增益常数 。设滤波器的脉冲响应为 $h(t)$，于是锁相环满足以下的方程：

$$v_c(t) = K_d\{\sin\omega_i t \cdot \cos[\omega_o t + \theta(t)]\} * h(t)$$

为了便于仿真，不妨采用一阶的 RC 低通滤波器，于是锁相环满足以下的二阶微分方程：

$$\frac{d^2\theta}{dt^2} + \frac{1}{RC}\cdot\frac{d\theta}{dt} = \frac{K_v \cdot K_d}{RC}\sin\omega_i t \cdot \cos[\omega_o t + \theta(t)]$$

我们知道，使用 *MATLAB* 对常微分方程进行数值解是很容易的。求出压控振荡器输出信号的瞬时相位后，便可以得到压控振荡器的输入电压 v_c（t）和压控振荡器输出的 瞬时频率 f（t）：

$$v_c(t) = \frac{1}{K_v}\cdot\frac{d\theta}{dt},\quad f(t) = f_o + \frac{1}{2\pi}\cdot\frac{d\theta}{dt}$$

自定义的 M 函数 pll.m 采用了求解微分方程的方法对模拟的锁相环进行仿真，其中使用的滤波器为一阶的 RC 低通滤波器，仿真过程结束之后，屏幕上显示出压控振荡器的输入电压、压控振荡器输出信号的波形和相位、压控振荡器输出的瞬时频率及频谱、锁相环的输入信号等参数，从中可以看出锁相环大致的工作过程。

pll.m 函数的用法是：

```
》pll (fi, fo, Kd, T, th0, dt);
```

CHROMINANCE VECTORSCOPE BBI

File Edit Window Help

COLOUR GRAPH — NTSC [C]: [.310 .316]

Violet:	390 - 450 nm
Blue:	450 - 500 nm
Green:	500 - 570 nm
Yellow:	570 - 590 nm
Orange:	590 - 610 nm
Red:	610 - 770 nm

xy: [0.384 0.319] HSV: [0 0.475 0.895]

Brown, Red, Green, Blue, Saturat, Value, Angle 103.5, R 0.75, Chrom Vector, Y: Contrast, Store, Compare, Restore, Graph/Vector, Info

图 6.14 色度三角形

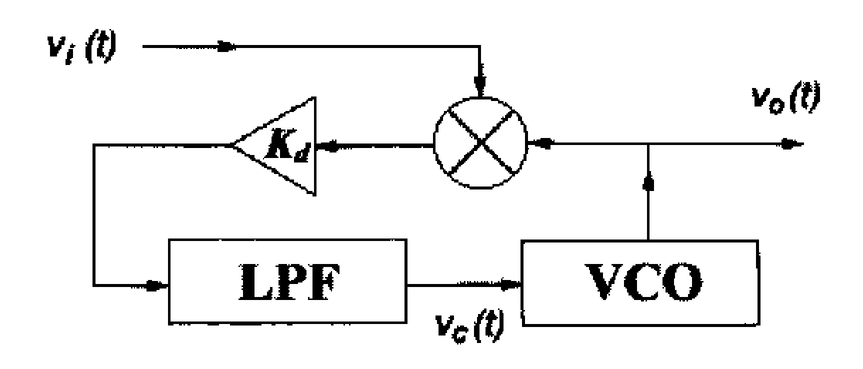


图 6.15 锁相环的模型

输入参数：

fi 输入信号频率（kHz），　　fo 压控振荡器的固有振荡频率（kHz）

Kd 鉴相器的增益常数，　　T 时间区间（s）

dt 时间间隔（s），　　th0＝［θ（0）θ′（0）］微分方程的初始条件

缺省的输入参数为：pll（800，1000，1，.1，［0 0］，.00025）；

pll.m 的程序清单如下：

```
function [tt, v1, fo, fs, t1, vc, th1, fn, c, vo] =pll (fii, foo, Kd, T, th0, dt);
%
%           [t1, vc, th1, fn, c, vo] =pll (fi, fo, Kd, T, th0, dt);
%           Default: pll (800, 1000, 1, .1, [0 0], .00025); BBI 99
glohal K fi fo
if nargin<5; th0= [0 0]; end;
if nargin<2; foo=1000; end;
if nargin<1; fii=800; end;
if nargin<3; Kd=1; end;
if nargin<4; T=100/foo; end;
if nargin<6; dt=.25/foo; end;
fo=foo; fi=fii; Kv=2*pi*fo; K=Kv*Kd;
vs=version; vs=vs (1);
if strcmp (vs,'4');
    [t, th] =ode23 ('equation', 0, T, th0); whitebg ('w'); delete (1);
else;
    dT=.25/fi; [t, th] =ode45 ('equation', 0: dT: T, th0');
end;
t1=0: dt: T-dt;
th1=spline (t, th (:, 1), t1); th2=spline (t, th (:, 2), t1);
vo=cos (2*pi*fo*t1+th1);
fn=1/T; n=length (vo); c=abs (fft (vo)) /n*2;
I=1: n/2; fn=fn* (I-1); c=c (I);
I=find (c>.01); fn=fn (I); cn=20*log10 (c (I));
vc=th2/Kv; f=th2/ (2*pi) +fo;
% ==============================================
nstr= ['Phase-Locked Loop (PLL) fo = ' num2str (fo) ' (kHz), Kv = '];
nstr= [nstr num2str (Kv) ' x 1e03 (rad/V-s), Kd =' num2str (Kd) ']; nstr= [nstr
'BBI 99'];
figure ('Units','normal','Pos', [0.006 .05 .985 .88]);
set (gcf,'NumberTitle','off','Name', nstr);
subplot (221); plot (t1, vc,'b'); title ('Control Voltage');
xlabel ('t ( ms )'); v=axis; v3=v (3) *1.17; zoom xon;
if abs (v3) <1e-06; v3= -0.084; end;
if abs (min (vc) +max (vc)) <1e-04; v3= -1.337; end;
subplot (223); plot (t1, f,'k');
title ('Frequency'); xlabel ('t ( ms )');
subplot (224); plot (t1, th1,'k');
```

```
title ('Phase'); xlabel ('t ( ms )');
m = length (I);
for i = 1: m;
     subplot (222);
     plot (fn (i) + [0 0], [-40 cn (i)], 'b'); hold on;
end;
plot (fn (m) +2/T, 0, 'b', fn (1) -2/T, 0, 'b');
title ('Spectrum'); hold off; xlabel ('f ( kHz )'); pause (3);
% ===================================================

m = 10 * 4; n = length (t1); n2 = n - m + 1;
dt = t1 (2); T = m * dt; dt = .1 * dt; fs = 1/dt;
tt = 0: dt: T; tt2 = t1 (n2) + tt;
v1 = spline (t1 (1: m), vc (1: m), tt);
v2 = spline (t1 (n2: n), vc (n2: n), tt2);
y1 - vco (v1, fo, fs); y2 = vco (v2, fo, fs);
nstr = ['SCOPE blue: vo (t) red: vi (t) fi = '];
nstr = [nstr num2str (fi) ' (kHz), fo = ' num2str (fo) ' (kHz),'];
H2 = figure ('Name', nstr, 'Num', 'off', 'Pos', [5 100 790 400]);
subplot (211); plot (tt, y1, 'b', tt, sin (2 * pi * fi * tt), 'r');
set (gca, 'units', 'pix', 'pos', [40 240 720 140]); xlabel ('t ( ms )');
subplot (212); plot (tt2, sin (2 * pi * fi * tt2), 'r', tt2, y2, 'b');
set (gca, 'units', 'pix', 'pos', [40 45 720 140]);
axis ( [tt2 (1) tt2 (length (tt2)) -1 1]);
xlabel ('t ( ms )'); v = axis; zoom xon;
```

pll.m 函数在运行的过程中要调用 equation.m 函数，该函数用来描述微分方程，其程序清单如下：

```
function yp = equation (x, y);
%
% BBI 99
global K fi fo
RC = 25/ (2 * pi * fo);          % 滤波器的时间常数
yp (1) = y (2);
yp (2) = K/RC * sin (2 * pi * fi * x) .* cos (2 * pi * fo * x + y (1)) - y (2) /RC;
v = version; v = v (1);
if strcmp (v, '5'); yp = yp'; end;
```

键入 pll (800, 1000, 1, .1, [0 0], .00025); 便可得到图 6.16 和图 6.17 中所示的结果，从图中可清楚地看出锁相环的大致工作过程。当锁相环进入了“锁定”状态之后，输出信号与输入信号的频率相同，而两者之间存在着一个恒定的相位差。

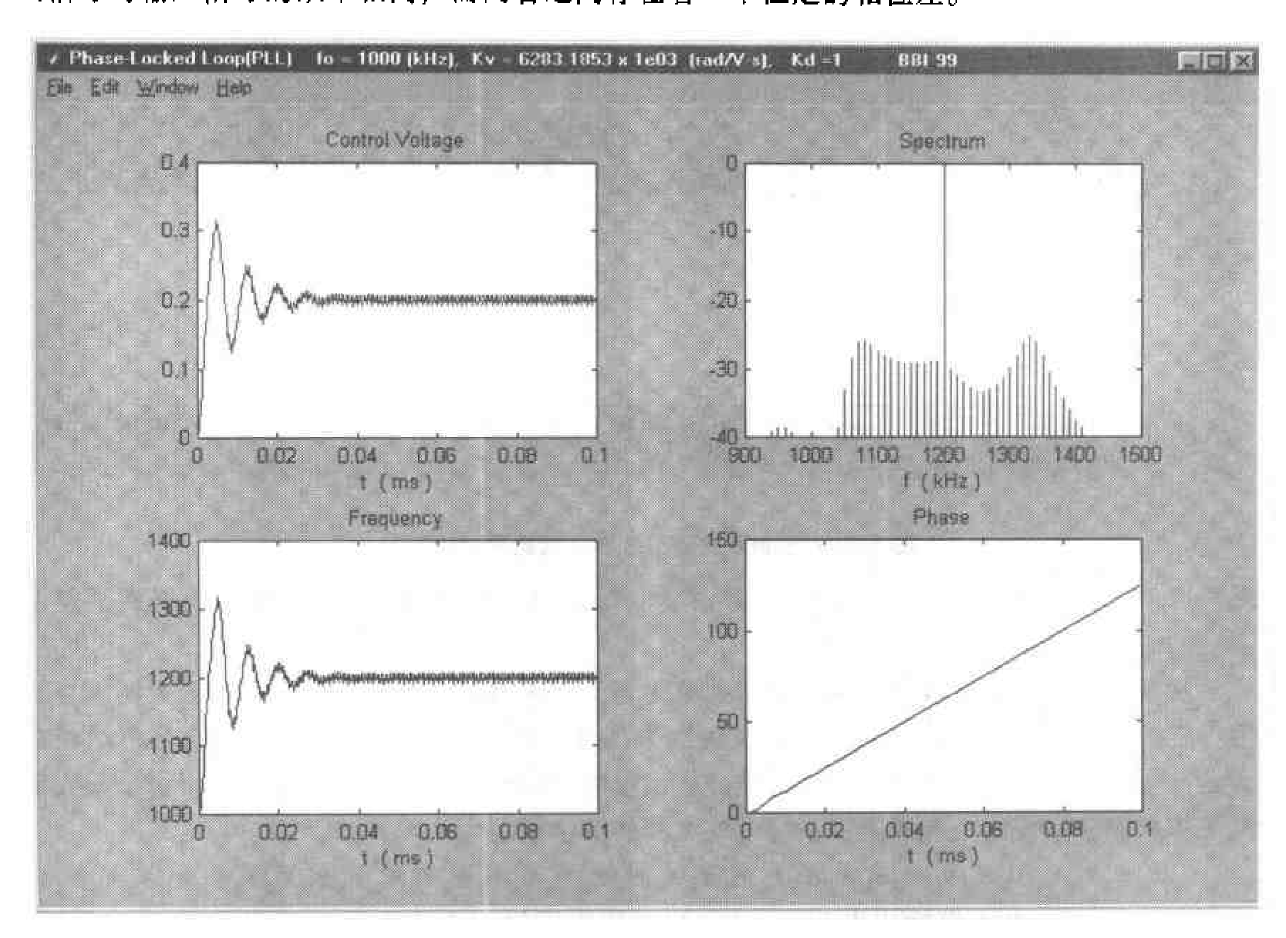


图 6.16 锁相环的控制电压、瞬时频率及频谱、相位

6.8 有线电视系统中 CTB 的测量

当一个有线电视系统内传输的电视频道数目超过 21 个以后，CTB（复合三次差 拍）就成为影响系统质量的主要因素。由于在理论上对系统内的 CTB 进行定量的分析几乎是不可能的，另外对一个有线电视系统的 CTB 指标进行测量也是一项比较复杂的工作，不但需要购置昂贵的测量设备，同时从事测量工作的技术人员必须要有相当的实际经验，因此在系统的设计过程中确有必要通过仿真来对 CTB 指标进行预测。

有线电视系统中 CTB 的测量方法是：将若干个信号源的输出信号混合在一起输入被测量的有线电视系统，每个信号源对应于一个电视频道，信号源的输出电平应按照“完全倾斜式”方式来设定，即低频端的输出电平要低于高频端的输出电平，这样与实际的系统工作状态是吻合的；记录某个电视频道图像载波处的电平，然后关闭该频道所使用的信号源，使用频谱仪在该频道的频率范围内进行测量，测出的最大频率分量与图像载波的电平进行比较，

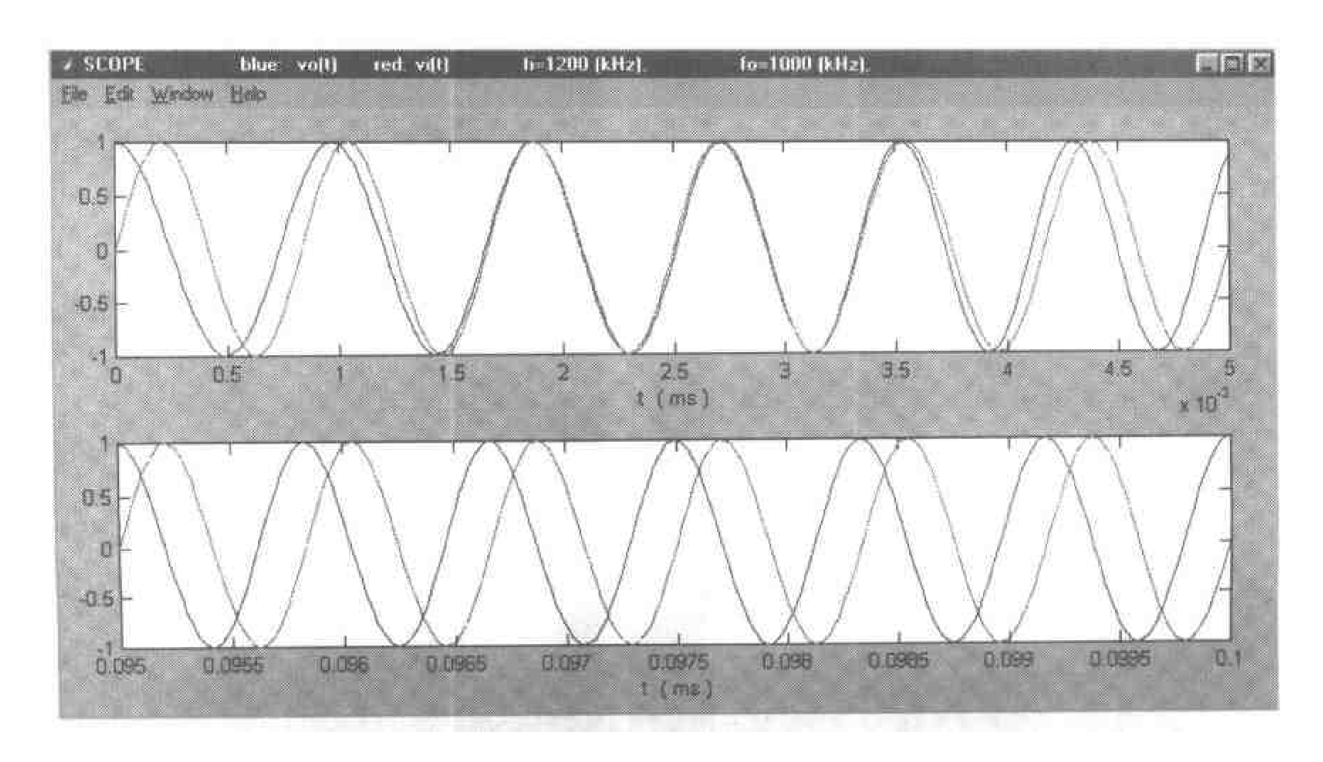


图 6.17 锁相环的输出波形（起始段、锁定段）

就可以得出上述频道处的 CTB 数值；对系统内每个频道都进行类似的测量，便得到了整个系统的 CTB 指标。

从理论讲，一个有线电视系统可以抽象为若干个串接的放大器，由于每个放大器都存在着一定的非线性失真，因此系统内的非线性失真（包括交调、CSO 和 CTB）就是由这些放大器产生的非线性失真组合而成，于是分析系统的非线性失真指标就是通过分析每个放大器的工作状态来完成的。有线电视系统内放大器的数学模型为：

$$u_o(t) = \alpha_1 \cdot u_i(t) + \alpha_2 \cdot u_i^2(t) + \alpha_3 \cdot u_i^3(t) \qquad (\mu V)$$

其中，$u_o(t)$ 为放大器输出电压，$u_i(t)$ 为放大器输入电压，α_1 为放大器的放大系数，它可以由放大器的增益求出，系数 α_2 代表了放大器的二次失真项，而系数 α_3 则代表了放大器的三次失真项，其量纲为 $1/(\mu V)^2$。

放大器的交调失真和 CTB 都是由三次失真项引起的，因此系数 α_3 是分析 CTB 问题的关键所在，它可以从德国工业标准 DIN45004B 中定义的放大器最大输出电平 U_{DIN} 得到。

$$\alpha_3 = 10^{-48/20} \cdot \frac{3 \cdot \alpha_1^3}{2 \cdot U_{DIN}^2}$$

设在有线电视系统中共有 N 个频道，每个频道的频率不同，信号电平不同，另外各频道的初始相位也不同。于是放大器输入信号的表达式为，

$$u_i(t) = u_m \sum_N \xi_n \sin(\omega_n t + \varphi_n)$$

从上式可以清楚地看出，放大器的输入信号是由各个不同强度、不同初相位的信号叠加而成，因此在分析放大器的非线性失真问题时，必须要同时考虑输入信号的强度和相位的影响。

放大器的非线性失真指标除了与放大器本身的特性参数有关之外，它还与系统内传输的

频道数目、频道的配置方式、放大器的实际输出电平等参数有关，其中对 CTB 指标影响最为显著的是放大器的实际输出电平，该电平又称为放大器的工作电平。当放大器的输出电平提高 1dB 时，其三阶失真项（交调、CTB）就提高 2dB，反之亦然。在有线电视系统的设计和调整过程中，最为关键的工作就是要给放大器选择一个比较合适的输出电平数值。考虑到在有线电视系统通常采用的"完全倾斜方式"，放大器的输出电平 U_{opt} 可以表示为：$U_{opt}=20\lg(\alpha_1 u_m)$，而每一个频道的输出电平与 U_{opt} 的关系为：$U_{opt}+20\lg\xi_i$。

将放大器输入信号的表达式代入放大器的数学模型之中，然后使用快速傅立叶变换（FFT）对放大器输出信号进行频谱分析，便可对 CTB 的测量进行仿真。

自定义的 M 函数 ctb.m、ctb1.m 和 ctb2.m 用来对单个放大器的 CTB 测量进行仿真，其中 ctb.m 的功能是确定有线电视系统的频道配置，设定各种参数，并且采用用户菜单界面的方式选择进行 CTB 测量的频道。ctb.m 函数的用法是：

》ctb（band，vout，vdin，slope）；

输入参数：

band 有线电视系统的带宽（MHz），vout 放大器的输出电平（dBμV）

vdin 放大器的最大输出电平（dBμV），Slope 斜率（dB）

缺省的输入方式为：》ctb（300，105，120，5）；

ctb.m 函数的程序清单如下：

```
function [] =ctb (band, vout, vdin, slope);
%
%   Usage: ctb (band, vout, vdin, slope); BBI 99
if nargin<4; slope=5; end;
if nargin<3; vdin=120; end;
if nargin<2; vout=105; end;
if nargin<1; band=300; end;
slope=abs (slope);
nStr= ['CATV CHANNELS BBI 99/11 Slope = ' num2str (slope) ' dB'];
nStr= [nStr ', fmax =' num2str (band) ' MHz' blanks (15)];
nStr= [nStr 'Vdin =' int2str (vdin) ' dBμV'];
figure ('Name', nStr, 'NumberTitle', 'off', ...
        'Position', [5 35 790 520]);
axes ('Position', [.07 .1 .91 .88]);
vs=version; vs=vs (1);
if strcmp (vs, '4'); whitebg ('k'); cstr='g'; ccstr='c';
else cstr='b'; ccstr='r';
end;
H2=uimenu ('Label', '&Auto-Test', 'Callback', 'ctb2;');
H1=uimenu ('Label', '&Channels');
     H11=uimenu (H1, 'Label', 'Ch.1-Ch.12');
```

```
for i=1: 12;
    ii=int2str (i);
    uimenu (H11,'Label', ['Ch.' ii],'Callback', ['ctb1 ' ii ';']);
end;
H12=uimenu (H1,'Label','Ch.13-Ch.24');
for i=13: 24;
    ii=int2str (i);
    uimenu (H12,'Label', ['Ch.' ii],'Callback', ['ctb1 ' ii ';']);
end;
H13=uimenu (H1,'Label','Ch.25-Ch.36');
for i=25: 36;
    ii=int2str (i);
    uimenu (H13,'Label', ['Ch.' ii],'Callback', ['cth1 ' ii ';']);
end;

H14=uimenu (H1,'Label','Ch.37-Ch.48');
for i=37: 48;
    ii=int2str (i);
    uimenu (H14,'Label', ['Ch.' ii],'Callback', ['ctb1 ' ii ';']);
end;

H15=uimenu (H1,'Label','Ch.49-Ch.60');
for i=49: 60;
    ii=int2str (i);
    uimenu (H15,'Label', ['Ch.' ii],'Callback', ['ctb1 ' ii ';']);
end;
H16=uimenu (H1,'Label','Ch.61-Ch.68');
for i=61: 68;
    ii=int2str (i);
    uimenu (H16,'Label', ['Ch.' ii],'Callback', ['ctb1 ' ii ';']);
end;
H21=uimenu (H1,'Label','Z1-Z12','Separator','on');
for i=101: 112;
    ii=int2str (i);
    uimenu (H21,'Label', ['Ch.' ii],'Callback', ['ctb1 ' ii ';']);
end;
H22=uimenu (H1,'Label','Z13-Z24');
for i=113: 124;
    ii=int2str (i);
```

```
        uimenu (H22,'Label', ['Ch.' ii],'Callback', ['ctb1 ' ii ';']);
    end;
    H23=uimenu (H1,'Label','Z25-Z37');
    for i=125: 137;
        ii=int2str (i);
        uimenu (H23,'Label', ['Ch.' ii],'Callback', ['ctb1 ' ii ';']);
    end;
uimenu (H1,'Label','Ch.24+1','Separator','on','Callback','ctb1 (201)');
uimenu (H1,'Label','Ch.24+2','Callback','ctb1 (202)');
uimenu (H1,'Label','Ch.24+3','Callback','ctb1 (203)');
uimenu (H1,'Label','Ch.24+4','Callback','ctb1 (204)');
uimenu (H1,'Label','Ch.24+5','Callback','ctb1 (205)');


% ==============================================================
fds=[49.75: 8: 65.75 77.25 85.25 168.25: 8: 216.25 471.25: 8: 559.25 607.25:
    8: 951.25];
fz= [112.25: 8: 161.25 224.25: 8: 456.25]; f24=567.25: 8: 599.25;
f=sort ( [fds fz f24]); I=find (f+6.75<=band);
f=f (I); f1=f (1); f2=max (f);
v=slope* (f-f1) / (f2-f1) +vout-slope;
phi=rand (size (f)) *2*pi;
% ==============================================================
plot ( [1 1] *f (1), [0 v (1)], cstr); hold on; %grid on;
for i=2: length (f);
    plot ( [1 1] *f (i), [0 v (i)], cstr);
end;
u=axis; axis ( [40 u (2) vout-70 vout+10]);
set (gca,'Xgrid','on');
hold off; zoom xon;
save ctbdata band vout vdin slope fds fz f24 f v phi cstr ccstr
CTB= []; CH= []; save ctbdata1 CTB CH
```

函数 ctb.m 在运行过程中要调用 ctb1.m 函数对某个单一频道来进行频谱分析，函数 ctb1.m 的程序清单如下：

```
function [] =ctb1 (n);
%            BBI 99
if isstr (n); n=str2num (n); end;
```

```
load ctbdata; load ctbdata1;
figure ('NumberTitle','off','Position', [5 35 790 520]);
axes ('Position', [.07 .1 .91 .88]);
if n<69;
    fch=fds (n); chStr= ['Ch.' int2str (n)];
elseif n<138;
    fch=fz (n-100); chStr= ['Z' int2str (n-100)];
elseif n>200;
    fch=f24 (n-200); chStr= ['Ch.24+' int2str (n-200)];
end;
J=find (f= =fch); I=find (f=fch); fs=2^nextpow2 (max (f) *6);
T=4; dt=1/fs; t=0: dt: T-dt; x=0; N=length (t); f1=1/T;
for i=1: length (f);
    x=x+10^ ( (v (i) -vout) /20) *sin (2*pi*f (i) *t+phi (i));
end;
x=x-10^ ( (v (J) -vout) /20) *sin (2*pi*fch*t+phi (J));
f=f (I); phi=phi (I);
% ==================================================================
x3=x.^3; y3=2*abs (fft (x3)) /N;
n=1: N/2; y3=y3 (n); f3=f1* (n-1);
I=find (f3> (fch-64) &f3< (fch+72)); y3=y3 (I); f3=f3 (I);
ctb=20*log10 (2/3) -48+2* (vout-vdin) +20*log10 (y3); % Formula
I=find (ctb>-80); ctb=ctb (I); f3=f3 (I); ctb1= [];
plot ( [1 1] *f3 (1), [vout-80 ctb (1) +vout], cstr); hold on;
for i=2: length (I);
    fi=f3 (i);
    if fi> (fch-1.25) &fi< (fch+6.75);
        lineC=ccstr; ctb1= [ctb1 ctb (i)];
    else;
        lineC=cstr;
    end;
    plot ( [1 1] *fi, [vout-80 cth (i) +vout], lineC);
end;
I=find (f>= (fch-64) &f<= (fch+72)); ff=f (I); vv=v (I);
for i=1: length (ff);
    plot ( [ff (i) ff (i)], [vout-80 vv (i)], [cstr ':']);
end;
V=axis; axis ( [V (1: 2) vout-80 vout+10]);
y=get (gca,'YTick'); n=length (y);
```

```
for i=1: n-1;
    H=plot (V (1: 2), [y (i) y (i)],':', V (1: 2), [y (i) y (i)] +5,':');
    set (H,'Color', [.4 .4 .2]);
end;
H=plot (V (1: 2), [y (n) y (n)],':'); set (H,'Color', [.4 .4 .2]);
hold off; zoom xon;
% ==========================================================
ctbm=max (ctb1+vout-v (J)); ctbm=round (ctbm*100) *0.01;
fchStr=num2str (rem (fch, 1));
fchStr= [num2str (fix (fch)) fchStr (2: 4) ' MHz'];
nameStr= ['CTB ANALYZER BBI' blanks (40) 'CTB = ' num2str (ctbm)];
nameStr= [nameStr ' dBc' blanks (10) chStr blanks (5) fchStr];
set (gcf,'Name', nameStr);
CTB= [CTB ctbm]; CH= [CH fch]; save ctbdata1 CTB CH
```

函数 ctb.m 在运行过程中还调用 ctb2.m 函数来对所有的频道自动进行频谱分析，函数 ctb2.m 的程序清单如下：

```
function [] =ctb2 ();
%                          BBI 99
figure ('Name','Auto Test, Please Wait','NumberTitle','off', ...
    'Pos', [120 80 560 420],'color','w');
axes ('Position', [.01 .01 .98 .98]);
set (gca,'color','w');
load ctbdata;
fs=2^nextpow2 (max (f) *6); m=length (f);
T=4; dt=1/fs; t=0: dt: T-dt;
N=length (t); f1=1/T; CTB= []; CH= []; X=0;
for i=1: m;
    X=X+10^ ( (v (i) -vout) /20) *sin (2*pi*f (i) *t+phi (i));
end;
for i=1: m;
    fch=f (i);
    x=X-10^ ( (v (i) -vout) /20) *sin (2*pi*f (i) *t+phi (i));
    x3=x.^3; y3=2*abs (fft (x3)) /N;
    n=1: N/2; y3=y3 (n); f3=f1* (n-1);
    I=find (f3> (fch-1.25) &f3< (fch+6.75)); y3=y3 (I); %f3=f3 (I);
    ctb=20*log10 (2/3) -48+2* (vout-vdin) +20*log10 (y3) +vout-v (i);
    CTB= [CTB max (ctb)]; CH= [CH fch];
```

```
end; CTB=round (CTB*100) *0.01;
%==========================================================
[CH, I] =sort (CH); CTB=CTB (I); n=length (CH); y=.98; m=0;
[ctbm, I] =max (CTB); chm=CH (I);
ch1=fix (chm); ch2=chm-ch1; ch2=num2str (ch2);
nameStr= ['RECORD OF MEASUREMENT SIMULATION' blanks (5)];
nameStr= [nameStr 'CTBm = ' num2str (ctbm) ' dBc ( '];
namestr= [namestr int2str (ch1) ch2 (2: 4) ' MHz)'];
set (gcf,'Name', nameStr);
fontC='kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk';
fontC= [fontC fontC fontC fontC fontC fontC];
fontC (I) ='r';
if n<10; i2=1; else; i2=ceil (n/10); end;
for i=1: i2;
    if i<i2; j2=10; else; j2=n- (i-1) *10; end;
    for j=1: j2;
        m=m+1; ch1=fix (CH (m)); ch2=CH (m) -ch1; ch2=num2str (ch2);
        text (.1*j, y, [int2str (ch1) ch2 (2: 4)],'FontSize', 9, ...
        'FontName','Arial','Hor','right','Color', fontC (m));
        text (.1*j, y-.03, num2str (CTB (m)),'FontSize', 9, ...
        'FontName','Arial','Hor','right','Color', fontC (m));
    end; y=y-0.08;
end; axis off;
```

图 6.18 为一个 300MHz 系统内各频道的电平分布，而图 6.19 和图 6.20 为该系统 CTB 指标的仿真测量结果。

6.9 模拟卫星电视接收系统

衡量一个模拟卫星电视接收系统的质量有若干个参数，其中最为重要的参数是载噪比(C/N 或 CNR)。模拟卫星电视接收系统载噪比的计算公式为

$$C/N = EIRP - L_s - \Delta L - \beta + G_{ANT} - 10\lg(T_A + T_{LNB}) - 10\lg(kB) \quad (dB)$$

其中，$EIRP$ 为卫星转发器的等效全向辐射功率，单位为 dBW；L_S 为电波在自由空间内的传播损耗；ΔL 为卫星链路的附加损耗；β 为上行链路因子；G_{ANT}为接收天线的增益；T_A 为接收天线的等效噪声温度；T_{LNB}为高频头的噪声温度；$k=1.38\times10^{-23}$为玻尔兹曼常数；B 为卫星接收机的中频带宽。

自定义的 M 函数 sattv.m 用来对模拟卫星电视链路进行计算，同时根据接收地点的位置和所接收的卫星位置计算出接收天线的仰角 EL、方位角 AZ 和极化角，并且对电视图像的质量进行大致的仿真。当载噪比大于卫星接收机的动态门限（12dB）时，图像上是没有噪点干扰的；反之，图像上就会出现噪点干扰；sattv.m 函数对这种情况也进行了仿真。

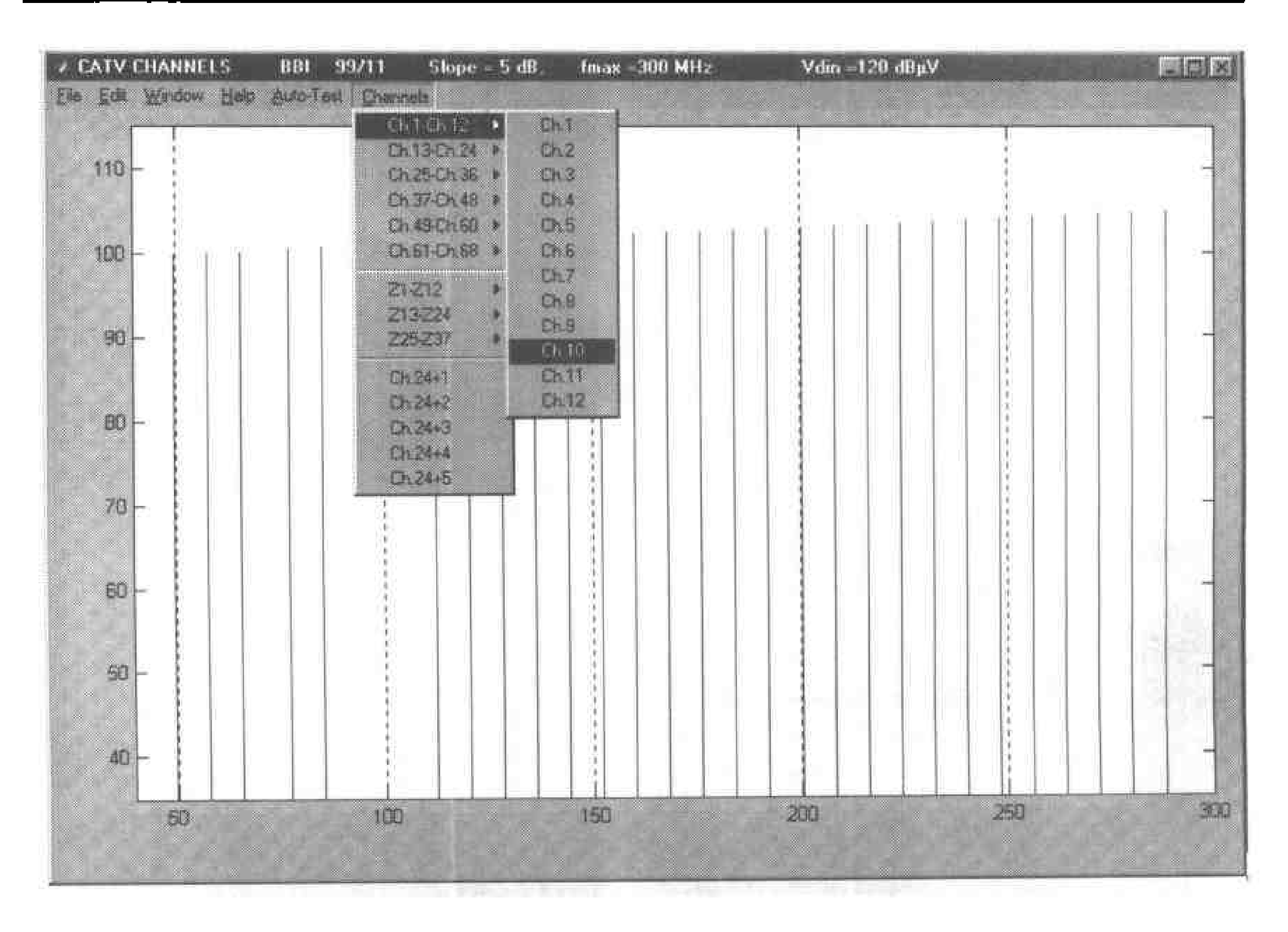


图 6.18 某 300MHz 系统各频道的电平

sattv.m 函数的用法是：

》[CNR, EL, AZ, alfa, R, G, GT, Pc] = sattv(lat, lnt, EIRP, lnts, band, d, e1, Tr, B, dL)；

输入参数：

lat	接收地点的纬度（°）,	lnt	接收地点的经度（°）
EIRP	卫星的等效全向辐射功率（dBW）,	lnts	卫星的经度（°）
band	波段（字符串 'C' 或 'Ku'）,	d	卫星接收天线的直径（m）
e1	卫星接收天线的口面效率,	Tr	高频头的噪声温度（K）
B	卫星接收机的中频带宽（Hz）,	dL	卫星链路的附加损耗（dB）

缺省的输入参数为：》sattv（39.9, 116.5, 35, 134, 'C', 4.5, 0.6, 30, 27, .5）;

输出参数：

CNR	系统的载噪比（dB）,	EL	接收天线的仰角（°）
AZ	接收天线的方位角（°）,	alfa	接收天线的极化角（°）
R	接收点与卫星之间的距离（km）,	G	卫星接收天线的增益（dB）
GT	卫星接收系统的优值（dB）,	Pc	接收天线输入功率（dBm）

sattv.m 函数的程序清单如下：

```
function [CNR, EL, AZ, alfa, R, G, GT, Pc] =sattv (lat, lnt, EIRP, ...
                                    lnts, band, d, e1, Tr, B, dL);
```

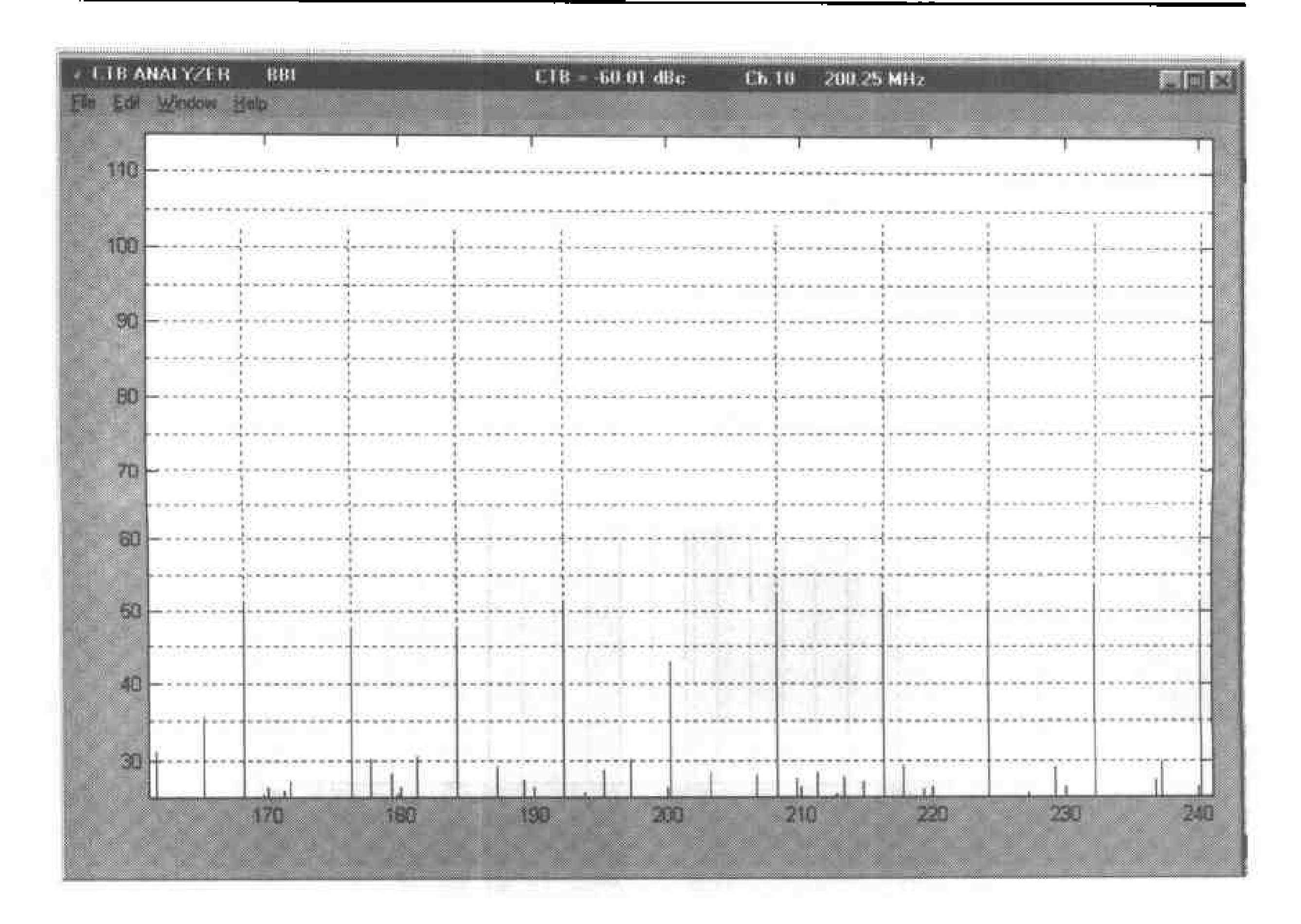


图 6.19 单个频道 CTB 的仿真测量结果（10 频道）

RECORD OF MEASUREMENT SIMULATION CTBm = -56.52 dBc (160.25 MHz)

File Edit Window Help

49.75 -58.29	57.75 -59.06	65.75 -60.67	77.25 -58.51	85.25 -65.05	112.25 -68.58	120.25 -63.57	128.25 -56.68	136.25 -65.6	144.25 -63.01
152.25 -67.7	160.25 -56.52	168.25 -61.42	176.25 -57.42	184.25 -64.75	192.25 -60.78	200.25 -60.01	208.25 -65.42	216.25 -58.15	224.25 -68.75
232.25 -60.83	240.25 -67.67	248.25 -61.53	256.25 -63.64	264.25 -70.83	272.25 -56.71	280.25 -60.41	288.25 -69.96		

图 6.20 CTB 的自动测量结果

```
%
%        BBI  2000
if nargin<1; lat=39.9; end;
if nargin<2; lnt=116.5; end;
if nargin<3; EIRP=35; end;
if nargin<4; lnts=134; end;
if nargin<5; band='C'; end;
```

```
if nargin<6; d=4.5; end;
if nargin<7; e1=.6; end;
if nargin<8; Tr=30; end;
if nargin<9; B=27; end;
if nargin<10; dL=0.5; end;
a=6.6108;
ph=(lnt-lnts)*pi/180;          costh=cos(ph)*cos(lat*pi/180);
AZ=tan(ph)/sin(lat*pi/180);    AZ=atan(AZ)*180/pi;
EL=(costh-1/a)/sqrt(1-costh^2)EL=atan(EL)*180/pi;
alfa=sin(ph)/tan(lat*pi/180);  alfa=atan(abs(alfa))*180/pi;
R=6378*sqrt(a*a+1-2*a*costh);
if strcmp(band,'C');
    lmd=.075; Ta=16+100*EL.^(-.75);
elseif strcmp(band,'Ku');
    lmd=.025; Ta=24+250*EL.^(-1);
end;
G=20*log10(pi*d/lmd)+10*log10(e1);
Ls=20*log10(4*pi*R*1000/lmd);
GT=G-10*log10(Ta+Tr);
CNR=EIRP-Ls-abs(dL)+GT-10*log10(1.38e-023*B*1e+06);
beta=.5; CNR=CNR-beta;          % beta - Noise Factor of Up-Link
Pc=EIRP-Ls-ahs(dL)+G+30;        % dBm
%-----------------------------------------
x=[ones(30,5) 2*ones(30,5) 3*ones(30,5) 4*ones(30,5)];
x=[x 5*ones(30,5) 6*ones(30,5) 7*ones(30,5) 8*ones(30,5)];
map=[1 1 1; 1 1 0; 0 1 1; 0 1 0; 1 0 1; 1 0 0; 0 0 1; 0 0 0; .4 .4 .4];
x1=ones(32,42)*9; x1(2:31,2:41)=x; x0=1;
image(x1); colormap(map); hold on;
set(gca,'xtick',[],'ytick',[]);
n=-160*CNR+1924; n=fix(n);
for i=1:n;
    plot([1 1.5]+rand(1,1)*38.5+x0,[1 1]+rand(1,1)*29+x0,'k');
    plot([1 1.5]+rand(1,1)*38.5+x0,[1 1]+rand(1,1)*29+x0,'w');
end; hold off;
cstr=['CNR = ' num2str(round(CNR*10)*.1) ' dB'];
elstr=['EL = ' num2str(round(EL*10)*.1) '°'];
azstr=['AZ = ' num2str(round(AZ*10)*.1) '°'];
apstr=['P_tilt = ' num2str(round(alfa*10)*.1) '°'];
nstr=[cstr ', ' elstr ', ' azstr ', ' apstr];
```

```
set (gcf,'num','off','name', ['Satallite Broadcast, ' nstr]);
title ( [band ' Band Transponder']);
```

图 6.21 为使用缺省参数时，函数 sattv.m 的运行结果。

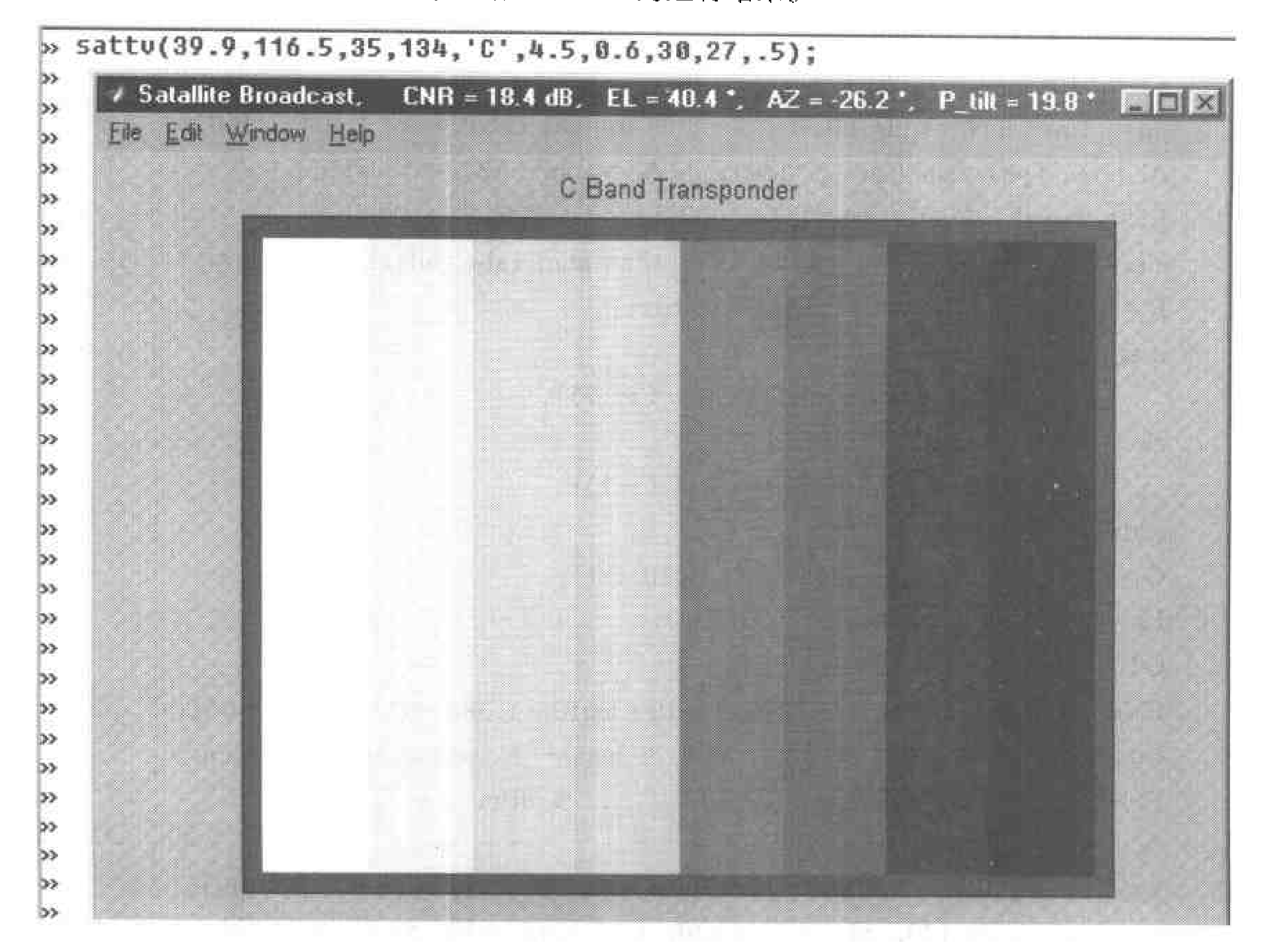


图 6.21 模拟卫星电视接收系统的仿真

参考文献

[1] 樊昌信，詹道庸，徐柄祥，吴成柯．通信原理（第四版）．北京：国防工业 出版社，1995

[2]（美）Leon W Couch. Digital and Analog Communication Systems. 北京：清华大学 出版社，1998

[3] 全子一，张家谋，刘诚．电视学原理．北京：国防工业出版社，1985

[4] Che Qing, Liu Jianbo. A Simulation Method for Analyzing the Intemodulation Distortion in a Cable TV System. Proceedings 5th International Symposium on Broadcasting Technology. Beijing 1997

[5] 龚汉民，闵士权．卫星广播技术．北京：北京广播学院出版社，1989

第七章　数字电子系统

在本章内通过列举实例的方式来介绍如何使用 MATLAB 对数字电子系统进行仿真的问题。这些实例均是由本书作者自己编写的，多数是使用 5.2 版 MATLAB 来进行的。

7.1　数字基带系统

从信息传输的角度来看，一个数字通信系统包括了两个重要的变换过程，一个信息与数字基带信号之间的变换，另一个是数字基带信号与信道信号之间的变换。在本节中对数字基带信号的一些问题进行仿真。

7.1.1　字符的编码与解码

本书中，字符的编码是采用 ASCII 码的方式将字符变换成二进制码，其中一个字节（8 比特）代表一个字符；而字符解码就是字符编码的逆过程。ASCII 码的全称是美国信息交换标准码，它出现在 1963 年，是目前计算机终端普遍使用的一种编码方式。

自定义的 M 函数 str2cod.m 用来实现字符串的编码过程，其用法是：

》 x = str2cod (s);　　s 为字符串，x 代表了二进制码流，它本身是一个行向量，其中每 8 位代表一个字符。

程序清单如下：

```
function x=str2cod (s);
%
%   Usage: x=str2cod (s);        BBI 2000
if nargin<1; s=' MATLAB 5.2/BBI 2000'; end;
   [m, n] =size (s);
for i=1: m
      xi=dec2bin (abs (s (i, :))) -48; [m1, n1] =size (xi);
      if n1==6;
         xi= [zeros (m1, 2) xi];
      elseif n1==7;
         xi= [zeros (m1, 1) xi];
      end;
         xi=xi'; xi=xi (:)'; x (i, :) =xi;
end;
```

自定义的 M 函数 cod2str.m 依照 ASCII 码将二进制的数据流变换成字符串，这样来实现解码的功能，显然 cod2str.m 和 str2cod.m 互为反函数。M 函数 cod2str.m 的用法是：

》s=cod2str (x)； s 为字符串，x 为二进制码流，它本身是一个行向量，其中每 8 位代表一个字符。

程序清单如下：

```
function y=cod2str (x);
%
%  Usage:   s=cod2str (x);    BBI  2000
%  to convert the binary vector to strings
if nargin<1;
  x=str2cod (str2mat (' MATLAB 5.2',' BBI 2000'));
end;
[m, n] =size (x); n=n/8;
for j=1: m;
  s= [];
  for i=1: n;
    I= (1: 8) + (i-1) *8;
    a=sum (x (j, I) .* [128 64 32 16 8 4 2 1]);
    s= [s setstr (a)];
    end;
    if j= =1; y=s; else; y=str2mat (y, s); end;
end;
```

键入 x=str2cod (' BBI')，就得到向量 [0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 1]，其中前 8 位的数值为 66，这就是大写字母 B 的 ASCII 码，最后 8 位的数值为 73，代表着大写字母 I。

接着键入 cod2str (x)，就得到了字符串 ' BBI'。

7.1.2 数字基带信号

数字基带信号是指信息代码的电压或电流波形，最常见的数字基带信号是用矩形电脉冲构成的，其具体形式有单极性不归零、双极性不归零、单极性归零、双极性归零、曼彻斯特码（双相码）等若干种。

自定义的 M 函数 pulse.m 根据二进制码流来形成矩形脉冲组成的数字基带信号，除了显示二进制码流和对应的数字基带信号之外，还显示出该信号的频谱，显然这对分析数字基带信号的特性是很有用的。pulse 函数的用法是：

》y=pulse (x, m, Kmod)； x 为代表二进制码流的向量；m=fs/fb，其中 fs 为采样频率，fb 为比特率，另外 m 要为整数；整数 Kmod 代表着数字基带信号的类型： 1－单极性不归零

2－双极性不归零　　　　　3－单极性归零

4－双极性归零　　　　　　5－曼彻斯特双极性

y为输出的数字基带信号，当函数未设输出参数时，屏幕上就显示数字基带信号的波形和频谱。

输入参数的缺省值为：x=randint (1, 32); m=16; Kmod=1;

程序清单如下：

```
function [y, m] =pulse (x, m, Kmod);
%
%   Usage: y=pulse (x, m, Kmod);        BBI 2000
if nargin<3; Kmod=1;                   end;
if nargin<2; m=16;                     end;
if nargin<1; x=randint (1, 32);        end;
di=.5;   Kmod=rem (Kmod, 5); Kmod= (Kmod==0) *5+Kmod;
if Kmod==1;
  xx= [x; x]; xx=xx (:)'; tstr=' Unipolor NRZ';
elseif Kmod==2;
  xx= [x; x]; xx=2*xx (:)' -1; tstr=' Polor NRZ';
elseif Kmod==3;
  xx= [x; zeros (size (x))]; xx=xx (:)'; di=.2;   tstr=' Unipolor RZ';
elseif Kmod==4;
  I=find (x==1); x1=x (I); x1=rem (cumsum (x1), 2) *2-1; xx=x;
  xx (I) =x1;
  xx= [xx; zeros (size (x))]; xx=xx (:)'; di=.2;   tstr=' Bipolor RZ';
elseif Kmod==5;
  xx= [x; rem (x+1, 2)]; xx=2*xx (:)' -1; tstr=' Manchester NRZ';
end;
y=xx;
for i=1: fix (m/2) -1;      y= [y; xx];   end; y=y (:)';
%-------------------------------------------------
if nargout<1;
  subplot (211); plot (1+ (1: length (y)) /m, y); hold on;
  v=axis; axis ( [v (1: 2) -1.2 1.5]); zoom xon;
  if Kmod==2|Kmod==0;
    plot ( [0 v (2)], [0 0],':');
  end; hold off;
  set (gcf,' num',' off',' name', ...
    [' Binary Pulse Generator, ' tstr ',' blanks (10) ...
     ' BBI 2000']);
```

```
    for i=1: length (x);
      text (i+di, 1.3, int2str (x (i)),' color',' b');
    end;
    title (' Pulse Signal');
    n=length (y); y=fft (y) /n; y=abs (y (1: fix (n/2))) *2;
    I=find (y<1e-04); y (I) =1e-04; y=20*log10 (y);
    f1=m/n; f=0: f1: (length (y) -1) *f1;
    subplot (212); plot (f, y,' r');
    title (' Power Spectrum'); xlabel (' f / fb');
end;
```

键入 pulse (str2cod (' MATLAB'), 16, 2); 便可得到字符串 ' MATLAB' 所对应的码流、双极性不归零脉冲构成的数字基带信号的波形和频谱，如图 7.1 所示，显然基带信号所占频率范围是相当宽的，因此在实际信道中传输起来是比较困难的。

在研究数字信号传输的过程中，对基带信号的频谱进行分析是必不可少的工作，由于数字基带信号为一随机脉冲序列，因此要对随机序列进行谱分析，这在理论上是可行的，但是也是相当复杂的。采用 MATLAB 仿真的方式，可以比较直观地给出信号序列与频谱的关系，这无疑对工程技术人员来说是很有帮助的。

7.1.3 微分编码

在通信信道中，很多设备会偶发“倒相”现象，使基带信号的波形反相，为了避免偶发的倒相现象诱发的误码，可以采用微分编码的方式。

设原码序列为 d_n，微分编码序列为 e_n，两者之间逻辑关系为异或关系，也可以表示为模 2 加为： $e_n=d_n \oplus e_{e-1}$ 　 $n = 1, 2, \cdots$

初始的参考相位有两种状态：$e_0=0$ 或 $e_0=1$。

微分译码的方式也为模 2 加：$d_n=e_n \oplus e_{n-1}$。

在 MATLAB 中完成微分编码可以使用累加后除 2 取余的方式，而微分解码则可直接采用 MATLAB 提供的异或函数 xor 来实现。自定义的 M 函数 dcode.m 即用来完成微分编码，其用法是：

》 y=dcode (x, rd); 　 x 为原码，y 为微分编码，rd 为初始参考相位（缺省值为 1）。

当函数未设定输出参数时，则自动显示原码和微分编码，这样便于进行对比。

程序清单如下：

```
function y=dcode (x, rd);
%
%   Usage: y=dcode (x);     BBI 2000
%   Differential Coding
if nargin<2; rd=1; end;
if nargin<1; x= [1 1 0 1 0 0 1 randint (1, 9)]; end;
```

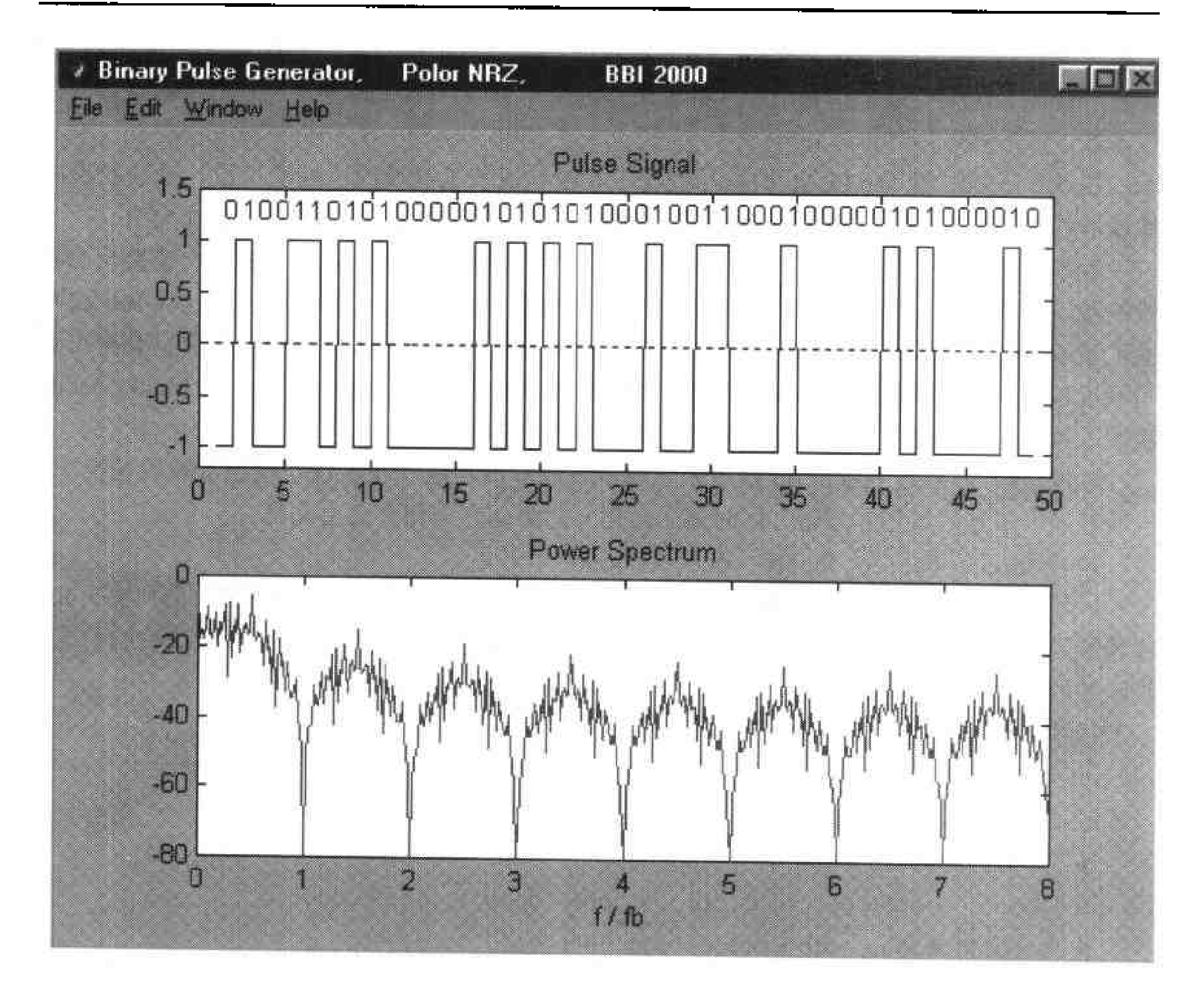


图 7.1 数字基带信号的波形与频谱

```
y=rem (cumsum (x), 2);                % Reference digit = 0
if rd==1;
  y=1-y;                              % Reference digit = 1
end;
if nargout<1; [x; y], end;
```

》dcode ([1 1 0 1 0 0 1 0 1 1 1], 0); 运行结果如下 (上面一行为原码):

1 1 0 1 0 0 1 0 1 1 1

1 0 0 1 1 1 0 0 1 0 1

微分译码的过程是十分简单的，只需一条命令即可完成：

x=xor (y, [rd y (1: length (y) -1)])，其中 rd 为初始参考相位，x 为译码输出。

7.1.4 基带成形

由矩形脉冲构成的数字基带信号所占的频带是十分宽的，从理论上讲，数字基带信号在频域内是无限延伸的。一个实际的信道，其带宽必定是有限的，在数字的传输过程中，有限

的频带必将会引起波形失真，从而产生码间（符号间）的干扰。为了解决码间干扰的问题，在数字传输系统中可以采用基带成形的方法，即使用升余弦滚降滤波器先对数字基带信号进行滤波，然后再进行传输。

在 5.2 版以上的 MATLAB 的信号处理工具箱和通信工具箱内，提供了设计升余弦滚降滤波器的 M 函数，故在对数字系统进行仿真时，最好采用 5.2 或 5.3 版的 MATALB。

为判断一个数字系统性能，通常采用“眼图”的方式。将示波器的 y 轴输入连接至滤波器的输出端，然后调整示波器的水平扫描时间，使其与信号的码元周期同步，这样在示波器上就显示出类似于人眼的波形，故称为眼图。有经验的工程技术人员从眼图中可以观察出码间干扰和噪声的影响程度，因此眼图是一种很实用的检验数字信道的方法。在 MATLAB 的通信工具箱内，提供了 eyescat.m 函数用来实现眼图的功能。

自定义的 M 函数 bshape.m 就用来完成基带成形的功能，在未设定输出参数的情况下，自动显示出信号的波形和频谱，同时显示出信号的眼图。该函数的用法是：

》 y=bshape (x, fs, fb, N, alfa, delay);

x 为数字基带信号，fs 为采样频率，fb 为比特率，N 为升余弦滚降滤波器的阶数，alfa 称为滚降系数，其取值范围在 0－1 之间，delay 代表眼图的观测延迟，y 为滤波后的信号。该函数的缺省数值为：fs=16，fb=1；N=16，alfa=0.5，delay=8。

程序清单如下：

```
function y=bshape (x, fs, fb, N, alfa, delay);
%
% Usge:    y=bshape (x, fs, fb, N, alfa, delay);    BBI 2000
% Baseband shaping with a raised cosine filter
if nargin<6; delay=8;          end;
if nargin<5; alfa=.5;          end;
if nargin<4; N=16;             end;
if nargin<3; fb=1;             end;
if nargin<2; fs=16;            end;
if nargin<1; x=pulse (randint (1, 64));    end;
b=firrcos (N, fb, 2*alfa*fb, fs);
y=filter (b, 1, x); y=y/max (abs (y)); m=fs;
%-----------------------------
if nargout<1;
   dt=1/m; t=0: dt: (length (y) -1) *dt;
   subplot (211); plot (t, x, t, y);
   v=axis; axis ( [v (1: 2) v (3) -v (4) *.2 v (4) *1.2]);
   set (gcf,' num',' off',' name', …
      [' Baseband Shaping with a Raised Cosine Filter…
      ( alfa = ' num2str (alfa) ' ),' blanks (10) ' BBI 2000']);
   title (' Pulse Signal');        h=get (gca,' children');
```

```
    set (h (1),' color',' b'); set (h (2),' color', [.9 .8 0]);
    subplot (224); n=length (y); eyescat (y (m: n), fb, fs, delay);
    y=fft (y) /n; y=abs (y (1: fix (n/4))) *2;
    I=find (y<1e-04); y (I) =1e-04; y=20*log10 (y);
    x=fft (x) /n; x=ahs (x (1: fix (n/4))) *2;
    I=find (x<1e-04); x (I) =1e-04; x=20*log10 (x);
    f1=m/n; f=0: f1: (length (y) -1) *f1;
    subplot (223); plot (f, x, f, y); h=get (gca,' children');
    set (h (1),' color',' b'); set (h (2),' color', [.9 .8 0]);
    title (' Power Spectrum'); xlabel (' f / fb'); zoom xon;
end;
```

键入 bshape；便得到使用缺省参数的情况下，随机的数字基带信号在滤波之前 和滤波之后的信号波形、频谱以及眼图，如图 7.2 所示。

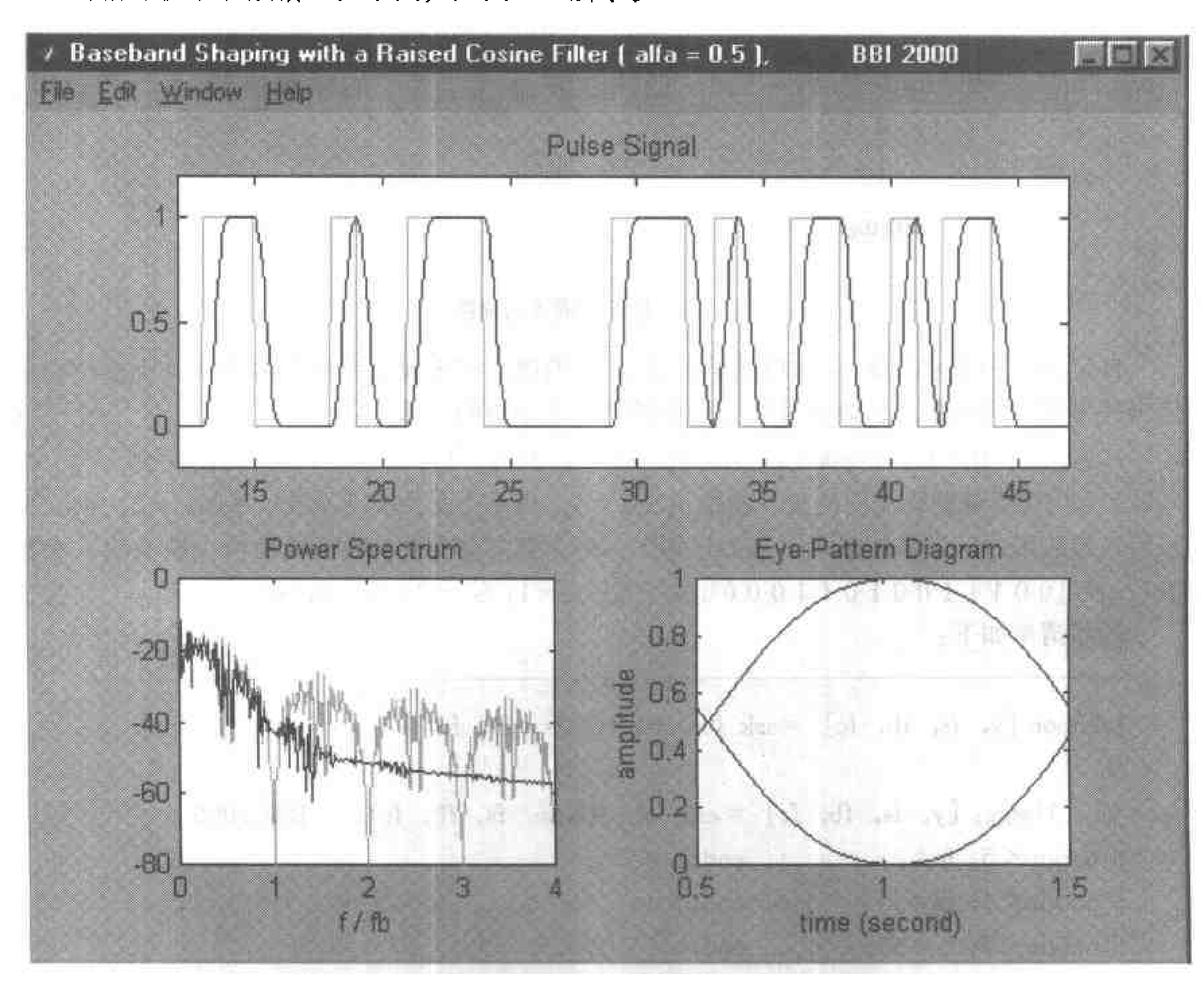


图 7.2　基带成形后的波形、频谱和眼图

7.2 二进制数字调制方式

与模拟调制类似，数字调制也有调幅、调频和调相三种基本方式。数字调制信号又称为键控信号，对于二进制数字调制来说，有振幅键控（ASK 或 OOK）、频移键控（FSK）、相移键控（BPSK）和微分相移键控（DPSK）等方式。

7.2.1 振幅键控（ASK）

振幅键控又称为开关键控（OOK），它类似于模拟调制方式中的调幅，属于线性调制的范畴，即已调波信号的频谱与基带信号的频谱结构相同。振幅键控的数学表达式为：

$$y(t) = \begin{cases} x(t) \cdot \sin\omega_c t, & x(t) > 0 \\ 0, & x(t) \leqslant 0 \end{cases}$$

其中，x 为双极性的基带信号，ω_c 为载波角频率。

振幅键控的解调方式有相干解调和不相干解调两种方式。相干解调器的框图见图 7.3，它由乘法器、低通滤波器和抽样判决器组成。

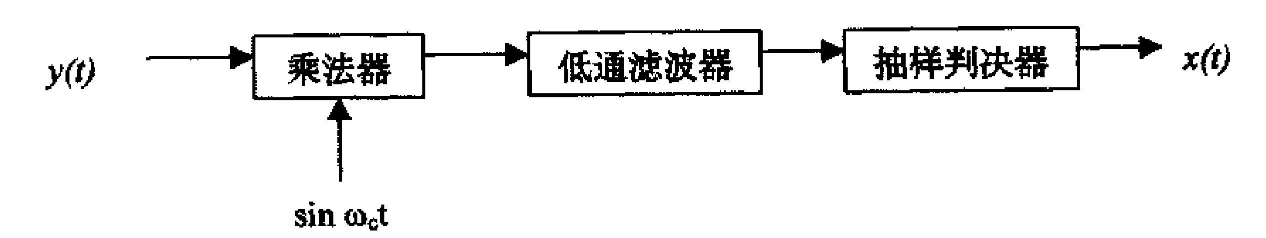


图 7.3 相干解调器的框图

自定义的 M 函数 ask.m 的功能是进行 ASK 调制，在未设定输出参数的情况下，自动显示出基带信号的波形，已调信号的波形和频谱。该函数的用法是：

》[y, fs, fb, fc] =ask (x, Kbase, fs, fb, fc);

x 为代表二进制码流的向量；参数 Kbase=1，表示不采用基带成形，Kbase=2，则表示采用基带成形；fs 为采样频率，fb 为比特率，fc 为载波频率。y 为已调信号。缺省输入参数为：x= [0 0 1 1 1 0 0 1 0 1 1 0 0 0 1 0]；Kbase=1；fs=32；fb=1；fc=2。

程序清单如下：

```
function [y, fs, fb, fc] =ask (x, Kbase, fs, fb, fc);
%
%   Usage: [y, fs, fb, fc] =ask (x, Kbase, fs, fb, fc);    BBI 2000
if nargin<5; fc=2;          end;
if nargin<4; fb=1;          end;
if nargin<3; fs=32;         end;
if nargin<2; Kbase=1;       end;
if nargin<1; x= [0 0 1 1 1 0 0 1 0 1 1 0 0 0 1 0]; end;
T=length (x) /fb;    m=fs/fb;
```

```
dt=1/fs;      t=0: dt: T-dt;
xx=pulse (x, m, 2);
if Kbase= =2;    xx=bshape (xx);    end;
y= (xx>=0) .* sin (2*pi*fc*t);    % ASK or OOK
%-------------------------------------
if nargout<1;
  subplot (211);    plot (t, y, t, xx*.5, [0 T], [0 0],' b:');
  title (' Signal');
  n=length (y); y=fft (y) /n; y=abs (y (1: fix (n/2))) *2;
  I=find (y<1e-04); y (I) =1e-04; y=20*log10 (y);
  f1=m/n; f=0: f1: (length (y) -1) *f1;
  subplot (212);    plot (f, y,' r');    grid on;
  title (' Spectrum');    xlabel (' f / fb'); zoom xon;
  set (gcf,' num',' off',' name', …
     [' ASK (OOK),' blanks (10) ' BBI 2000']);
end;
```

自定义的 M 函数 askdet.m 采用相干方式进行 ASK 解调，在未设定输出参数的情况下，自动显示出已调信号的波形、低通滤波器的输出波形、抽样判决器中经过整形的脉冲信号波形和解调出的二进制码流。该函数的用法是：

》[xn, x] =askdet (y, fs, fb, fc);

y 为 ASK 调制信号，fs 为采样频率，fb 为比特率，fc 为载波频率。xn 为解调输出的二进制码流向量，x 为解调输出的脉冲信号波形。

程序清单如下：

```
function [xn, x] =askdet (y, fs, fb, fc);
%
%  Usage: [xn, x] =askdet (y, fs, fb, fc); BBI 2000
if nargin<4; fc=2.0;      end;
if nargin<3; fb=1;        end;
if nargin<2; fs=32;       end;
if nargin<1; y=ask (str2cod (' CH'));    end;
dt=1/fs; n=length (y); t= (0: n-1) *dt;
[b, a] =butter (4, 2*fb/fs);
x=y. *sin (2*pi*fc*t);    x=filtfilt (b, a, x);
x1=max (x); x2=min (x); x=x- (x1+x2) *.5;
m=fs/fb;    N=n/m; n= (.75: 1: N) *m;
xn= (sign (x (n)) +1) /2;
%----------------------------
```

```
if nargout<1;
  c=' bbbbbbbbrrrrrrrr';
  subplot (211); plot (t, y);      title (' Input');
  subplot (212); plot (t, x, t, sign (x) *.3);
  set (gca,' ygrid',' on');    v=axis;
  for i=1: N;
    ci=rem (i, 16); ci=ci+ (ci==0) *16; ci=c (ci);
    text ( (2*i-1) *m*dt/2, v (4) *.8, int2str (xn (i)), …
      ' color', ci,' hor',' center');
  end;   title (' Output');
  set (gcf,' num',' off',' name', [' Coherent Detection of ASK (OOK),' …
    blanks (10) ' BBI 2000']);
end;   zoom xon;
```

》ask (str2cod (' MATLAB 5.2'), 1, 32, 1, 3); ASK 调制的结果如图 7.4 所示。

》y=ask (str2cod (' MATLAB 5.2'), 1, 32, 1, 3); askdet (y, 32, 1, 3);

ASK 解调的结果如图 7.5 所示。

7.2.2 频移键控（FSK）

频移键控为一种实用的数字调制方式，它采用两个不同频率的载波，一个代表 1，另一个代表 0，其数学表达式为：

$$y(t)=\begin{cases}x(t)\cdot\sin\omega_{c1}t, & x(t)>0\\ x(t)\cdot\sin\omega_{c2}t, & x(t)\leqslant 0\end{cases}$$

其中，x 为双极性的基带信号，ω_{c1}和 ω_{c2} 为两个载波角频率。

频移键控的解调有多种方式，其中相干解调器的结构如图 7.6 所示，它由乘法器、带通滤波器、低通滤波器、抽样判决器和加法器组成。

自定义的 M 函数 fsk.m 的功能是进行 FSK 调制，在未设定输出参数的情况下，自动显示出基带信号的波形，已调信号的波形和频谱。该函数的用法是：

》[y, fs, fb, fc1, fc2] =fsk (x, Kbase, fs, fb, fc1, fc2);

x 为代表二进制码流的向量；参数 Kbase=1，表示不采用基带成形，Kbase=2，则表示采用基带成形；fs 为采样频率，fb 为比特率，fc1 和 fc2 为载波频率。y 为已调信号。缺省输入参数为：x= [0 0 1 1 1 0 0 1 0 1 1 0 0 0 1 0]; Kbase=1; fs=32; fb=1; fc1=2; fc2=2.5
。

程序清单如下：

```
function [y, fs, fb, fc1, fc2] =fsk (x, Kbase, fs, fb, fc1, fc2);
%
%  Usage: [y, fs, fb, fc1, fc2] =fsk (x, Kbase, fs, fb, fc1, fc2);    BBI 2000
if nargin<6; fc2=2.5;     end;
```

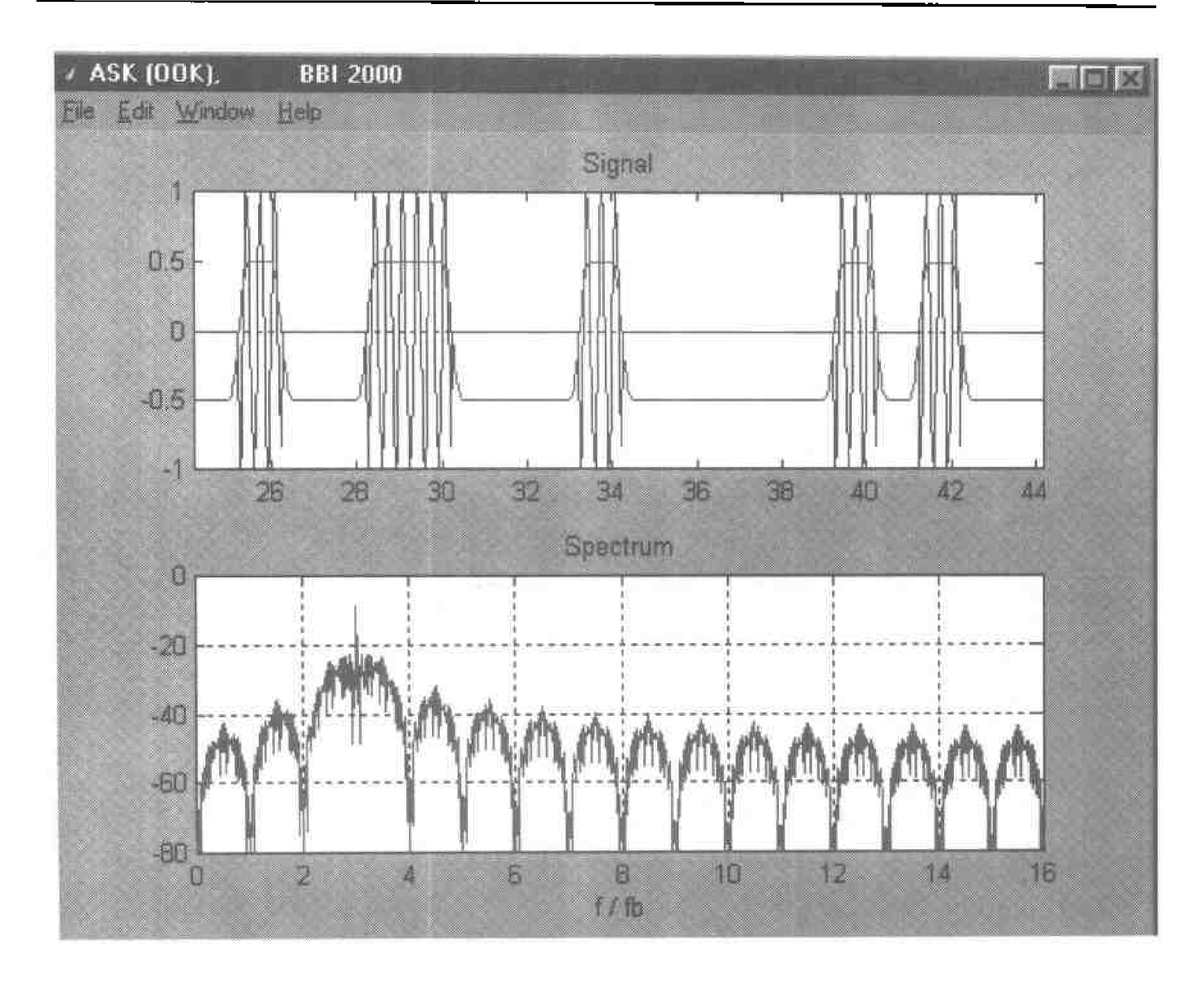


图 7.4 ASK 调制的波形和频谱

```
if nargin<5; fc1 =2;          end;
if nargin<4; fb=1;            end;
if nargin<3; fs=32;           end;
if nargin<2; Kbase=1;         end;
if nargin<1; x= [0 0 1 1 1 0 0 1 0 1 1 0 0 0 1 0]; end;
T=length (x) /fb;             m=fs/fb;
dt=1/fs;                      t=0: dt: T-dt;
xx=pulse (x, m, 2);
if Kbase= =2;      xx=bshape (xx);     end;
        % Discontinuous-Phase FSK
y= (xx>=0) .* sin (2 * pi * fc1 * t) + (xx<0) .* sin (2 * pi * fc2 * t);
% ---------------------------------------------
if nargout<1;
  subplot (211); plot (t, y, t, xx * .5, [0 T], [0 0],' b:');
  title (' Signal');
```

图 7.5　ASK 解调

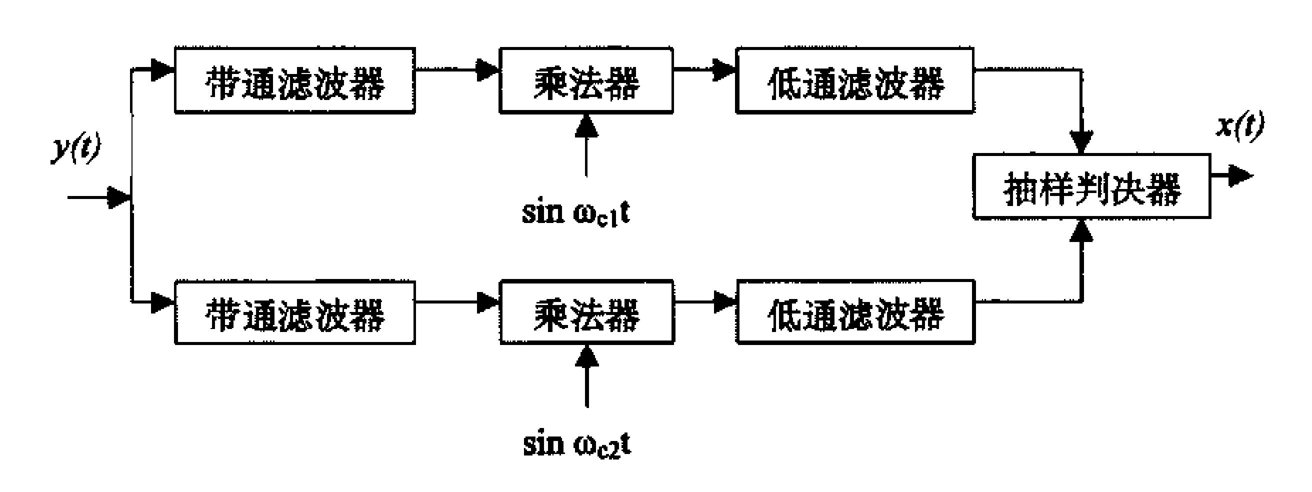


图 7.6　FSK 相干解调器的框图

```
n=length (y); y=fft (y) /n; y=abs (y (1: fix (n/2))) *2;
I=find (y<1e-04); y (I) =1e-04; y=20*log10 (y);
f1=m/n; f=0: f1: (length (y) -1) *f1;
```

```
  subplot (212); plot (f, y,' r'); grid on;
  title (' Spectrum'); xlabel (' f / fb'); zoom xon;
  set (gcf,' num',' off',' name', [' Discontinuous - Phase FSK,' …
  blanks (10) ' BBI 2000']);
end;
```

自定义的 M 函数 fskdet.m 采用相干方式进行 FSK 解调，在未设定输出参数的情况下，自动显示出已调信号的波形、低通滤波器的输出波形、抽样判决器中经过整形的脉冲信号波形和解调出的二进制码流。该函数的用法是：

》[xn, x] =fskdet (y, fs, fb, fc1, fc2);

y 为 FSK 调制信号，fs 为采样频率，fb 为比特率，fc1 和 fc2 为载波频率。xn 为解调输出的二进制码流向量，x 为解调输出的脉冲信号波形。

程序清单如下：

```
function [xn, x] =fskdet (y, fs, fb, fc1, fc2);
%
%  Usage: [xn, x] =fskdet (y, fs, fb, fc1, fc2); BBI 2000
if nargin<5; fc2=2.5;          end;
if nargin<4; fc1=2.0;          end;
if nargin<3; fb=1;             end;
if nargin<2; fs=32;            end;
if nargin<1; y=fsk (str2cod (' CH'));   end;
dt=1/fs; n=length (y); t= (0: n-1) * dt;
[b, a] =butter (4, 2*fb/fs);
[b1, a1] =cheby1 (3, .5, 2*fc1/fs);
y1=filtfilt (b1, a1, y) .*sin (2*pi*fc1*t);
y1=filtfilt (b, a, y1);
[b2, a2] =cheby1 (3, .5, 2.5*fc1/fs,' high');
y2=filtfilt (b2, a2, y) .*sin (2*pi*fc2*t);
y2=filtfilt (b, a, y2);
m=fs/fb;   N=n/m; n= (.75: 1: N) *m;
x=y1-y2;   xn= (sign (x (n)) +1) /2;
%-------------------------------------------------
if nargout<1;
  c=' bbbbbbbbrrrrrrrr';
  subplot (211); plot (t, y); title (' Input');
  subplot (212); plot (t, x, t, sign (x) *.5);
  set (gca,' ygrid',' on'); v=axis;
  for i=1: N;
```

```
    ci = rem (i, 16); ci = ci + (ci = = 0) * 16; ci = c (ci);
    text ( (2 * i - 1) * m * dt/2, v (4) * .8, int2str (xn (i)), ...
      ' color', ci, ' hor', ' center');
  end;   title (' Output');
  set (gcf, ' num', ' off', ' name', [' Coherent Detection of FSK ' ...
  blanks (10) ' BBI 2000']);
end;   zoom xon;
```

》fsk (str2cod (' BBI 2000'), 2);　　FSK 调制的结果如图 7.7 所示。

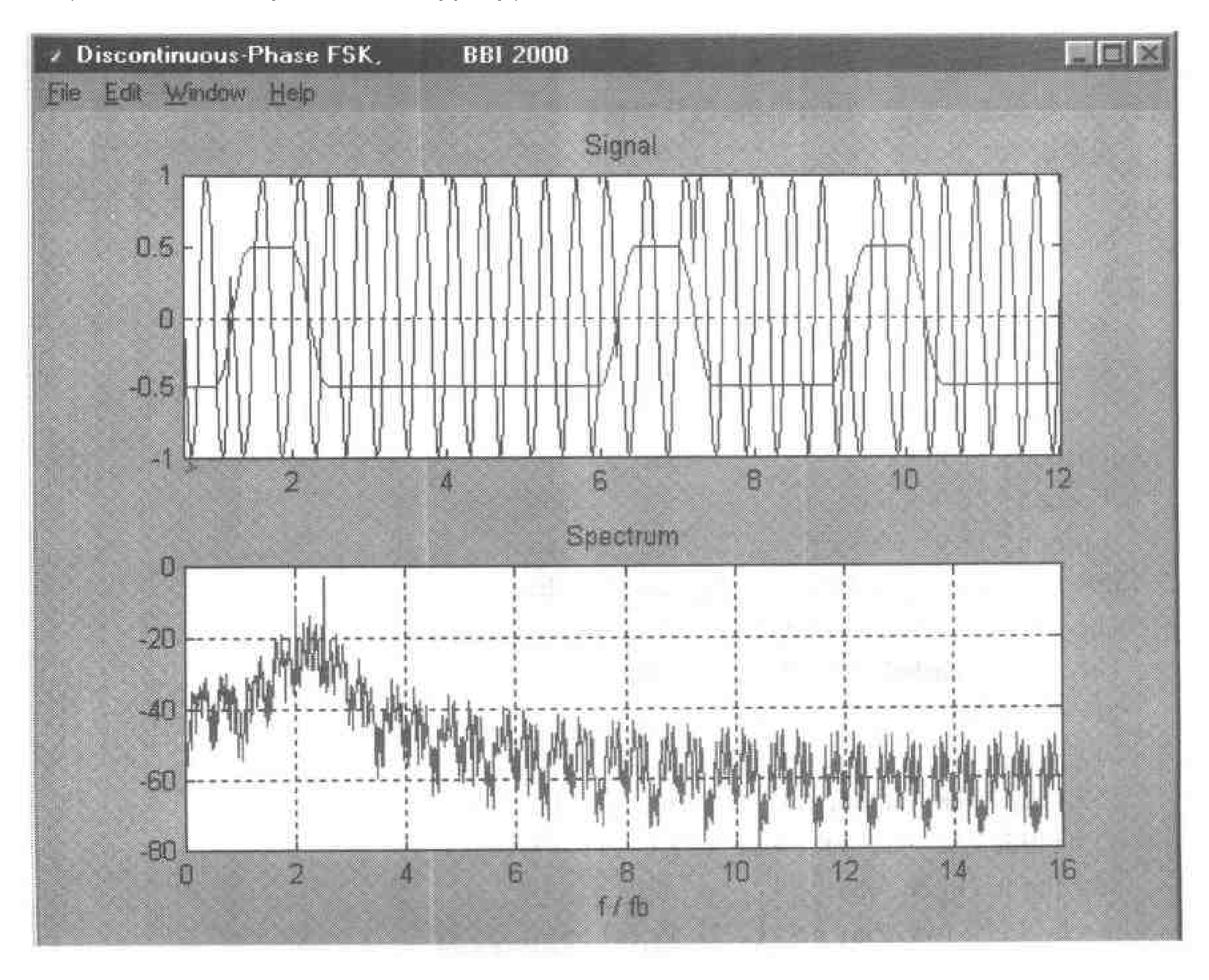


图 7.7　FSK 调制的波形和频谱

》y = fsk (str2cod (' BBI 2000'), 2); fskdet (y);　　FSK 解调的结果如图 7.8 所示。

7.2.3　相移键控 (BPSK) 与微分相移键控 (DPSK)

相移键控 (BPSK) 为一种性能较好的二进制数字调制方式，它采用两个了两个不同的相位，一个代表 1，另一个代表 0，其数学表达式为：

$$y(t) = x(t) \cdot \sin(\omega_c t)$$

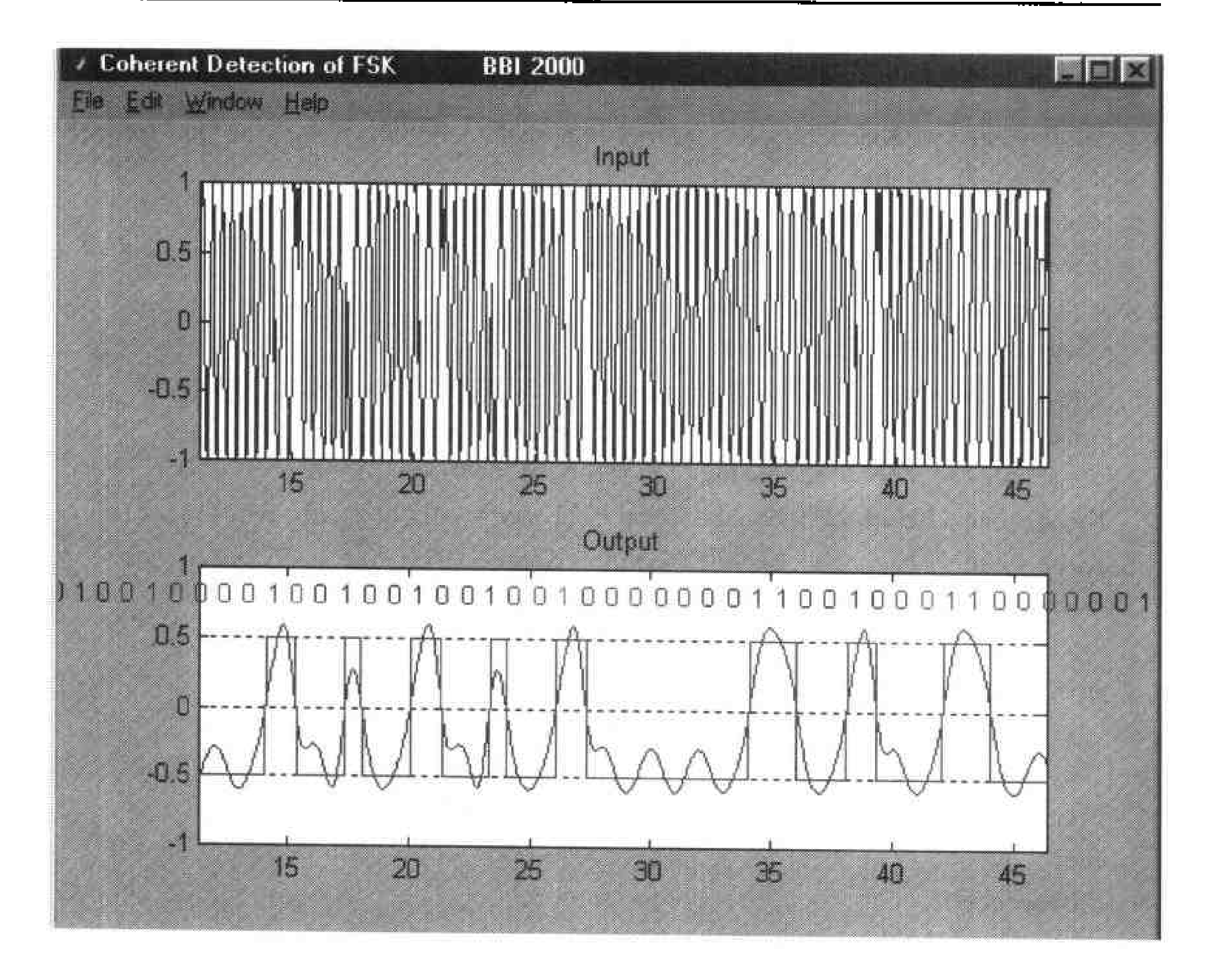


图 7.8 FSK 解调

其中，x 为双极性的基带信号，ω_c 为载波角频率。

采用微分编码的相移键控就称为微分相移键控（DPSK），从调制的角度来看，相移键控和微分相移键控是没有什么区别的。

相移键控的解调有多种方式，其中相干解调器的结构如图 7.3 所示，它由乘法器、低通滤波器和抽样判决器组成。

自定义的 M 函数 bpsk.m 的功能是进行 BPSK 和 DPSK 调制，在未设定输出参数的情况下，自动显示出基带信号的波形，已调信号的波形和频谱，以及信号的相位图。该函数的用法是：

》[y，Kmod，fs，fb，fc，rd] =bpsk（x，Kmod，Kbase，fs，fb，fc，rd）；

x 为代表二进制码流的向量；参数 Kmod=1，表示 BPSK 调制，Kmod=2，表示 DPSK 调制；参数 Kbase=1，表示不采用基带成形，Kbase=2，则表示采用基带成形；fs 为采样频率，fb 为比特率，fc 为载波频率，rd 为微分编码的初始参考相位。y 为已调信号。缺省输入参数为：x= [0 0 1 1 1 0 0 1 0 1 1 0 0 0 1 0]；Kbase=1；fs=32；fb=1；fc=2；rd=0 。

程序清单如下：

```
function [y, Kmod, fs, fb, fc, rd] =bpsk (x, Kmod, Kbase, fs, fb, fc, rd);
%
%  Usage: [y, Kmod, fs, fb, fc, rd] =bpsk (x, Kmod, Kbase, fs, fb, fc, rd);
if nargin<7; rd=0;          end;
if nargin<6; fc=2;          end;
if nargin<5; fb=1;          end;
if nargin<4; fs=32;         end;
if nargin<3; Kbase=1;       end;
if nargin<2; Kmod=1; end;    % Kmod=1, 2PSK;    Kmod=2, 2DPSK;
if nargin<1; x= [0 0 1 1 1 0 0 1 0 1 1 0 0 0 1 0]; end;
T=length (x) /fb;    m=fs/fb;    dt=1/fs;    t=0: dt: T-dt;
Kmod=rem (Kmod, 2); Kmod=Kmod+ (Kmod==0) *2; rd=rem (rd, 2);
Kbase=rem (Kbase, 2); Kbase=Kbase+ (Kbase==0) *2;
if Kmod==1;
  x1=1-x*2;        tstr=' BPSK';
elseif Kmod==2;
  x1=dcode (x, rd);    % rd - reference digit
  x1=1-rem (x1, 2) *2;
  tstr= [' DPSK (reference digit: ' int2str (rd) ')'];
end;
if Kbase==1;
  y=pulse (x1, m) .*sin (2*pi*fc*t);
elseif Kbase==2;
  x2=pulse (x1, m); x2=bshape (x2); %, fs, fb, N, alfa);
  y=x2.*sin (2*pi*fc*t);
end;
I=find (abs (y) <1e-09 & [diff (y) 0] >0);
mm=fs/fc;       I=rem (I, mm);       I=I+ (I==0) *mm;
ph= (m-I+1) *2*pi/mm;       ph=rem (pb, 2*pi);
%-------------------------
if nargout<1;
  subplot (211);
  if Kbase==1;
    plot ( [t T], [y y (1)], t, pulse (x, m) -.5, [0 T], [0 0],' b:');
  elseif Kbase==2;
      plot ( [t T], [y y (1)], t, -x2*.5, [0 T], [0 0],' b:');
      tstr= [tstr ' with tbe RC filter'];
  end;    title (' Signal');
  n=length (y); y=fft (y) /n; y=abs (y (1: fix (n/2))) *2;
```

```
  I=find (y<1e-04); y (I) =1e-04; y=20*log10 (y);
  f1=m/n; f=0: f1: (length (y) -1) *f1;
  subplot (223);      plot (f, y,' r');      grid on;
  title (' Spectrum');    xlabel (' f / fb');    zoom xon;
  subplot (224); H=polar (ph, ones (size (ph)),' ro');
  set (H,' markers', 8,' linew', 2);       xlabel (' Phase');
  set (gcf,' num',' off',' name', [tstr ',' blanks (10) ' BBI 2000']);
end;
```

自定义的M函数bpskdet.m采用相干方式进行BPSK和DPSK解调，在未设定输出参数的情况下，自动显示出已调信号的波形、低通滤波器的输出波形、抽样判决器中经过整形的脉冲信号波形和解调出的二进制码流。该函数的用法是：

》[xn, x] =bpskdet (y, Kmod, fs, fb, fc, rd);

y为PSK调制信号，参数Kmod=1，表示BPSK解调，Kmod=2，表示DPSK解调；fs为采样频率，fb为比特率，fc为载波频率。xn为解调输出的二进制码流向量，x为解调输出的脉冲信号波形。

程序清单如下：

```
function [xn, x] =bpskdet (y, Kmod, fs, fb, fc, rd);
%
%  Usage: [xn, x] =bpskdet (y, Kmod, fs, fb, fc, rd);   BBI 2000
if nargin<6; rd=0;          end;
if nargin<5; fc=2;          end;
if nargin<4; fb=1;          end;
if nargin<3; fs=32;         end;
if nargin<2; Kmod=1;        end;
if nargin<1; y=bpsk ( [0 0 1 1 1 0 0 1 0 1]); end;
dt=1/fs; t=0: dt: (length (y) -1) *dt;
x=y.*sin (2*pi*fc*t);
[b, a] =butter (2, 2*fb/fs);
x=-filtfilt (b, a, x);
m=fs/fb;    N=length (y) /m; n= (.75: 1: N) *m;
xn= (sign (x (n)) +1) /2;
if Kmod==1;
  tstr=' BPSK,';
elseif Kmod==2;
  xn=xor (xn, [rd xn (1: N-1)]); tstr=' DPSK,';
end;
%-----------------------------------
```

```
if nargout<1;
  c=' bbbbbbbbrrrrrrrr';
  subplot (211); plot (t, y);    title (' Input');
  subplot (212); plot (t, x, t, sign (x)  *.5);
  set (gca,' ygrid',' on');       v=axis;
  for i=1: N;
    ci=rem (i, 16); ci=ci+ (ci==0)  *16; ci=c (ci);
    text ( (2*i-1)  *m*dt/2, v (4)  *.8, int2str (xn (i)),' color', ci);
  end;    title (' Output');
  set (gcf,' num',' off',' name', [' Coherent Detection of ' …
  tstr blanks (10) ' BBI 2000']);     zoom xon;
end;
```

例:》bpsk (str2cod (' BBI 2000'), 1, 2); BPSK 调制的结果如图 7.9 所示。

》y=bpsk (str2cod (' BBI 2000'), 1, 2); bpskdet (y); BPSK 解调的 结果如图 7.10 所示。

7.3 多进制数字调制方式

多进制数字调制方式采用了多进制的数字基带信号去调制载波的频率、相位和幅度，与二进制数字调制方式相比，其主要特点是频谱利用率高，也就是说在同一频带内可以传输较多的信息，故多进制数字调制又称为高效率的数字调制方式。目前，在数字通信、数字微波中继、数字卫星广播、数字视频广播、数字音频广播等领域中，广泛地采用了多进制的数字调制方式，如 QPSK、QAM 等。

7.3.1 QPSK 与 QDPSK

QPSK 的含义是“四相相移键控”，而 QDPSK 的含义是“四相相对相移键控”，顾名思义，这两种数字调制都采用了调相的方式。QPSK 调制的码元为相位，在相平面一共有四个相位（即四个码元），每个相位（码元）代表着 2 比特。由于 QPSK 调制对信道载噪比的要求是比较低的，因此在数字卫星广播和数字微波中继系统普遍地采用了 QPSK 调制。

图 7.11 为 QPSK 调制器的原理框图。首先，串并行变换器将输入的二进制码流序列变换成为两行并列的码流序列（一个记为 I，另一个记为 Q），I、Q 序列的速率为输入序列速率的一半，其波形均为双极性不归零脉冲。通过基带成形处理后，I、Q 信号分别与两个相位相差 90°的载波信号相乘，然后相加就形成了四相相移键控信号，四个相位是 45°、135°、225°和 315°。由于信道中的噪声干扰主要作用在信号的幅度上，而对相位产生的影响比较小，因此 QPSK 调制多用于信号较弱的通信链路和广播链路中，如卫星广播系统、数字微波中继系统等。

QPSK 调制的数学表达式为：

$$y(t) = I(t) \cdot \cos(\omega_c t) - Q(t)\sin(\omega_c t)$$

QDPSK 调制的工作原理与 QPSK 调制基本相同，只是在串并行变换器之前增设了码元

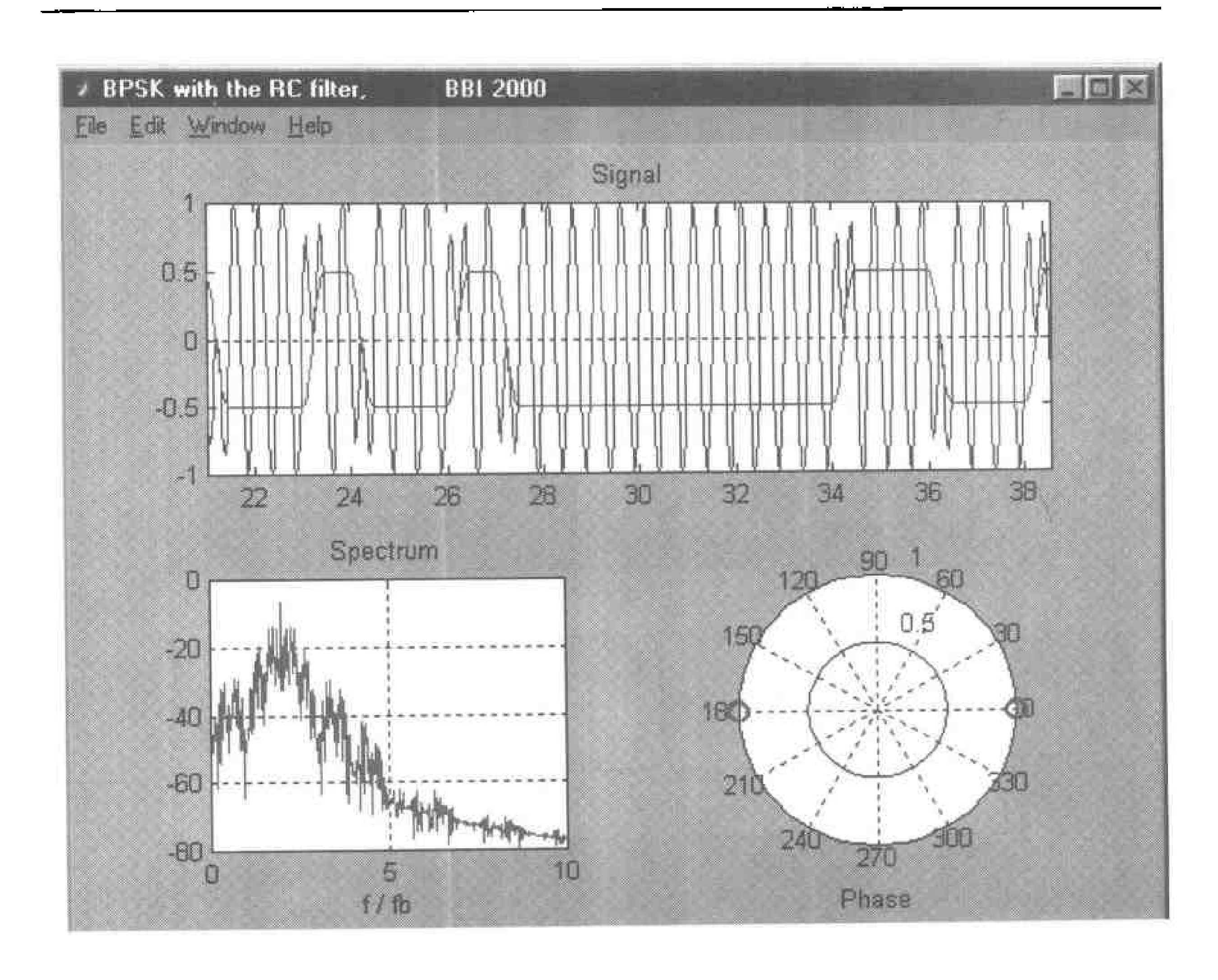


图 7.9　BPSK 调制

变换器，双比特的原码 an bn 与变换码 cn dn 之间的逻辑关系见表 7.1。

表 7.1　QDPSK 码变换表

	a_n　b_n		
c_{n-1}　d_{n-1}			
0　0	1　0	1　1	0　1
0　0	0　0	1　0	1　1
0　1	1　0	1　0	1　1
0　1	0　0	1　1	1　1
0　1	0　0	1　0	0　1
0　1	0　0	1　0	1　1

QPSK 和 QDPSK 的解调亦可采用相干解调的方式，如图 7.12 所示。而对于 QDPSK 解调，只需在串并行变换器后设置一个码变换器。

自定义的 M 函数 qpsk.m 的功能是进行 QPSK 和 QDPSK 调制，在未设定输出参数的情

图 7.10　BPSK 解调

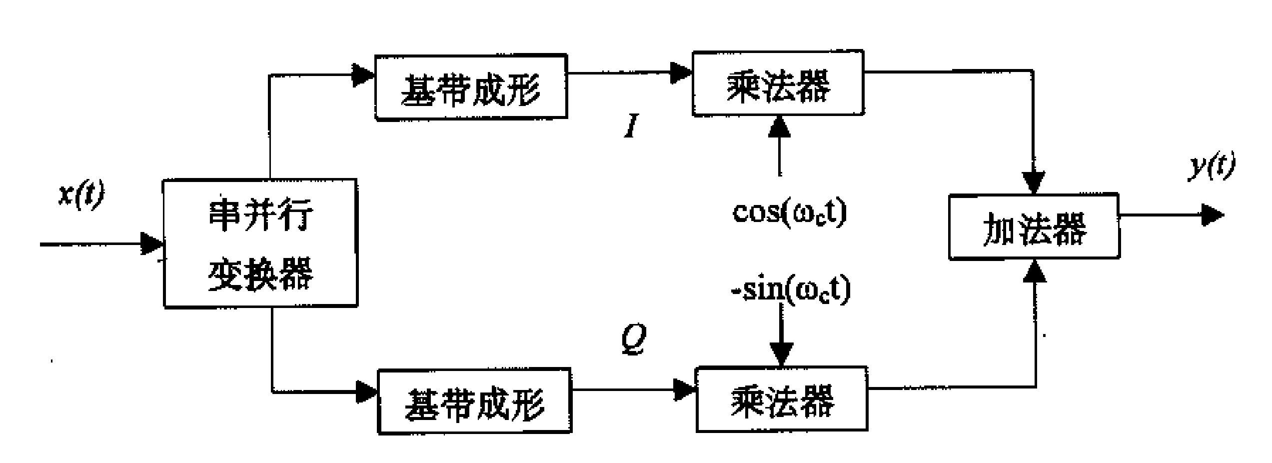


图 7.11　QPSK 调制器原理框图

况下，自动显示出基带信号的波形，已调信号的波形和频谱，以及信号的星座图。该函数的用法是：

》[y，Kmod，fs，fb，fc] =qpsk（x，Kmod，Kbase，fs，fb，fc)；

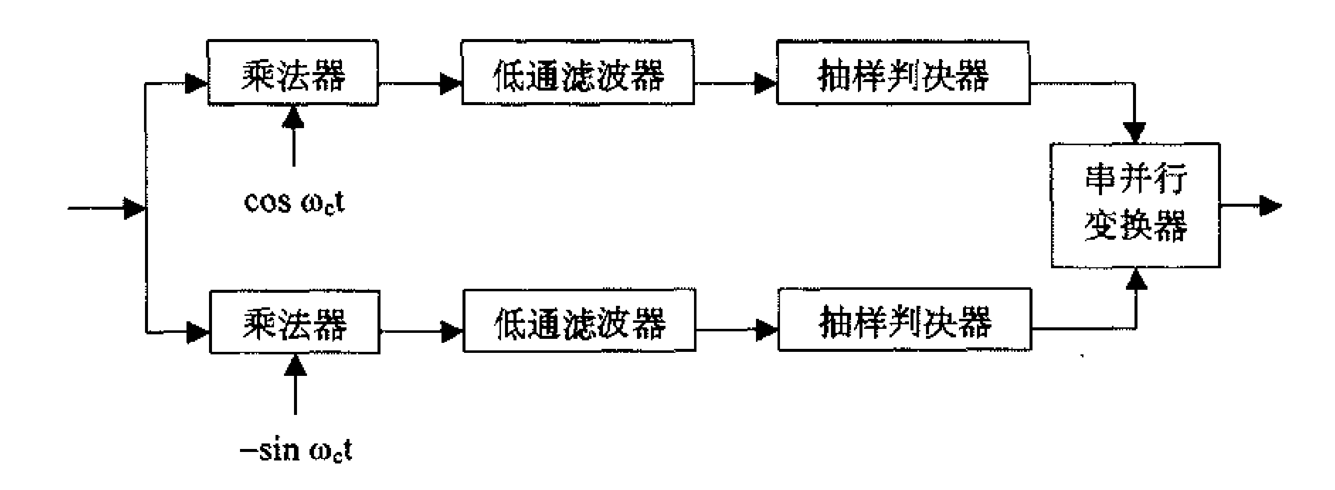


图 7.12　QPSK 相干解调器原理框图

x 为代表二进制码流的向量；参数 Kmod＝1，表示 QPSK 调制，Kmod＝2，为 QDPSK 调制；参数 Kbase＝1，表示不采用基带成形，Kbase＝2，则表示采用基带成形；fs 为采样频率，fb 为比特率，fc 为载波频率，y 为已调信号。

缺省输入参数为：x＝［0 0 0 1 1 1 1 0］；Kbase＝2；fs＝32；fb＝1；fc＝2。

程序清单如下：

```
function [y, Kmod, fs, fb, fc] =qpsk (x, Kmod, Kbase, fs, fb, fc);
%
%  Usage: [y, Kmod, fs, fb, fc] =qpsk (x, Kmod, Kbase, fs, fb, fc); BBI 2000
if nargin<6; fc=2;          end;
if nargin<5; fb=1;          end;
if nargin<4; fs=32;         end;
if nargin<3; Kbase=2;       end;
if nargin<2; Kmod=1;        end;    % Kmod=1, QPSK;    Kmod=2, QDPSK;
if nargin<1; x= [0 0 0 1 1 1 1 0]; end;
T=length (x) /fb;       m=fs/fb;     nn=length (x);
dt=1/fs;    t=0: dt: T-dt;
Kmod=rem (Kmod, 2);              Kmod=Kmod+ (Kmod==0) *2;
Kbase=rem (Kbase, 2);            Kbase=Kbase+ (Kbase==0) *2;
if Kmod==1;
   x1=x*2-1;        tstr=' QPSK';
elseif Kmod==2;
   x1=dqpskcd (x);       x1=x1*2-1;       tstr= [' QDPSK'];
end;
I=x1 (1: 2: nn-1);    Q=x1 (2: 2: nn);
I=pulse (I, 2*m);    Q=pulse (Q, 2*m);
if Kbase==2;
```

```
  I=bshape (I, fs, fb/2);      Q=bshape (Q, fs, fb/2);
end;
y=I. * cos (2 * pi * fc * t) -Q. * sin (2 * pi * fc * t);    y1=y;
n=length (y);
%-----------------------------------------
if nargout<1;
  subplot (211);
  plot (t, y, t, I, t, Q, [0 T], [0 0],' b:');
if Kbase= =2;
    tstr= [tstr ' with the RC filter'];
  end;     title (' Signal');
  n=length (y); y=fft (y) /n; y=abs (y (1: fix (n/2))) *2;
  I=find (y<1e-04); y (I) =1e-04; y=20 * log10 (y);
  f1=m/n; f=0: f1: (length (y) -1) * f1;
  subplot (223);     plot (f, y,' r');     grid on;
  title (' Spectrum');   xlabel (' f / fb');   zoom xon;
  subplot (224);   constel (y1, fs, fb, fc); xlabel (' Constellation');
  set (gcf,' num',' off',' name', [tstr ',' blanks (10) ' BBI 2000']);
end;
```

M 函数 qpsk.m 在运行过程中调用 qdpskcd.m 函数来完成 QDPSK 的码变换功能，调用 constel.m 函数来绘制信号的星座图。

M 函数 qdpskcd.m 的程序清单如下：

```
function y=qdpskcd (x, Kmod);
%
% Code & Decode for QDPSK  BBI 2000
if nargin<2; Kmod=1; end;
if nargin<1; x=str2cod (' MATLAB'); end;
T= [0 3; 1 2];
n=length (x);    y (1: 2) =x (1: 2);    a=T (y (1) +1, y (2) +1);
for i=3: 2: n-1;
  xn=x (i: i+1) +1; b=T (xn (1), xn (2));
  if Kmod= =1;       % code
     [y (i), y (i+1)] =find (T= =rem (b+a, 4));
     a=T (y (i), y (i+1));
  elseif Kmod= =2;      % decode
     if b<a; b=b+4; end;
      [y (i), y (i+1)] =find (T= =rem (b-a, 4));
```

```
    a=rem (b, 4); a=a+ (a==0) *4;
  end;
  y (i: i+1) =y (i: i+1) -1;
end;
```

M 函数 constel.m 的程序清单如下：

```
function c=constel (x, fs, fb, fc);
%
% Usage: ph=phdet (x, fs, fc);    BBI 2000
if nargin<4; fc=2;          end;
if nargin<3; fb=1;          end;
if nargin<2; fs=32;         end;
if nargin<1;
  t= (0: 255) /fs; x=cos (2*pi*fc*t+.25*pi);
end;
N=length (x);    m=2*fs/fb;       n=fs/fc;
i1=m-n;       i=1;    pb0= (i1-1) *2*pi/n;
while i <= N/m;
  xi=x (i1: i1+n-1);    y=2*fft (xi) /n; c (i) =y (2);
  i=i+1;    i1=i1+m;
end;
%------------------------------------
if nargout<1;
  cmax=max (abs (c));
  ph= (0: 5: 360) *pi/180; plot (1.414*cos (ph), 1.414*sin (pb),' c');
  hold on;
  for i=1: length (c);
    ph=angle (c (i)) -ph0+pi; % +pi/2;
    a=abs (c (i)) /cmax*1.414;
    plot (a*cos (ph), a*sin (ph),' r*');
  end;
  plot ( [-1.5 1.5], [0 0],' k:', [0 0], [-1.5 1.5],' k:');
  hold off;   axis equal;   axis ( [-1.5 1.5 -1.5 1.5]);
end;
```

自定义的 M 函数 qpskdet.m 采用相干方式进行 QPSK 和 QDPSK 解调，在未设定输出参数的情况下，自动显示出已调信号的波形、低通滤波器的输出波形、抽样判决器中经过整形的脉冲信号波形和解调后输出的二进制码流。该函数的用法是：

》[xn, x] =qpskdet (y, Kmod, fs, fb, fc);

y 为调制信号，参数 Kmod=1，表示 QPSK 解调，Kmod=2，表示 QDPSK 解调；fs 为采样频率，fb 为比特率，fc 为载波频率。x_n 为解调输出的二进制码流向量，x 为解调输出的脉冲信号波形。

程序清单如下：

```
function [xn, x] =qpskdet (y, Kmod, fs, fb, fc);
%
%   Usage: [xn, x] =qpskdet (y, Kmod, fs, fb, fc);        BBI 2000
%   Detection of QPSK and QDPSK
if nargin<5; fc=2;          end;
if nargin<4; fb=1;          end;
if nargin<3; fs=32;         end;
if nargin<2; Kmod=1;        end;
if nargin<1; y=qpsk ( [0 0 1 1 1 0 0 1 0 1]); end;
Kmod=rem (Kmod, 2);        Kmod=Kmod+ (Kmod==0) *2;
dt=1/fs; t=0: dt: (length (y) -1) *dt;
I=y.*cos (2*pi*fc*t); Q=-y.*sin (2*pi*fc*t);
[b, a] =butter (2, 2*fb/fs);
I=filtfilt (b, a, I);      Q=filtfilt (b, a, Q);
m=2*fs/fb; N=length (y) /m; n= (.75: 1: N) *m;
In= (sign (I (n)) +1) /2;       Qn= (sign (Q (n)) +1) /2;
xn= [In; Qn]; xn=xn (:);    xn=xn';    x= [I; Q];
if Kmod==1;
   tstr=' QPSK,';
elseif Kmod==2;
   xn=dqpskcd (xn, 2); tstr=' QDPSK,';
end;
%------------------------------------------------------------
if nargout<1;
   c=' bbbbbbbbrrrrrrrr';
   subplot (211); plot (t, y); title (' Input');
   subplot (212); plot (t, x); set (gca,' ygrid',' on'); v=axis;
   for i=1: 2*N;
      ci=rem (i, 16); ci=ci+ (ci==0) *16; ci=c (ci);
      text ( (2*i-1) *m*dt/4, v (4) *.8, int2str (xn (i)),' color', ci);
   end;    title (' Output');
   set (gcf,' num',' off',' name', ...
       [' Coherent Detection of ' tstr blanks (10) ' BBI 2000']);
```

```
zoom xon;
end;
```

例:》qpsk (str2cod (' MATLAB 5.2'), 1, 2);　　QPSK 调制的结果如图 7.13 所示。

》y=qpsk (str2cod (' MATLAB 5.2'), 1, 2); qpskdet (y); QPSK 解调的 结果如图 7.14 所示。

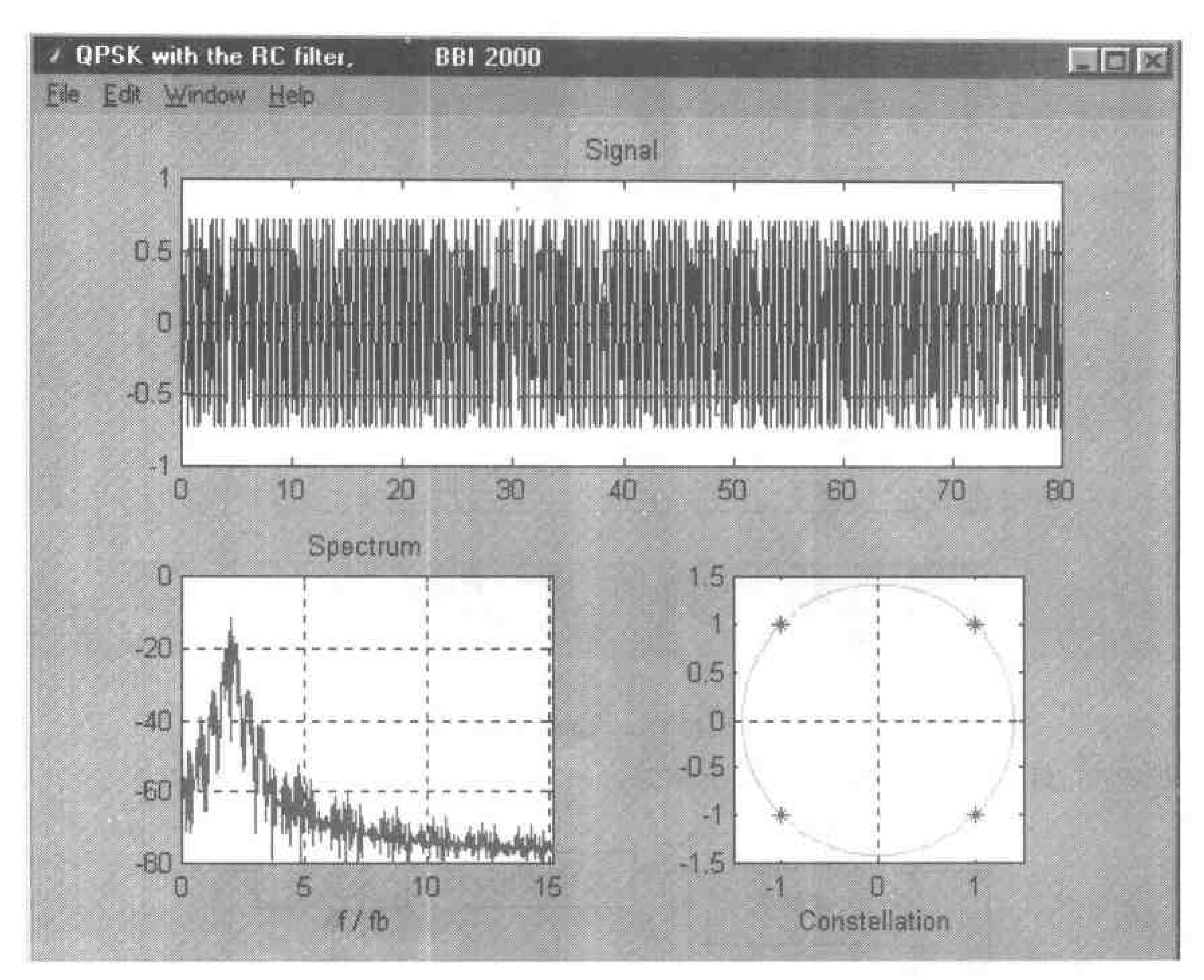


图 7.13　QPSK 调制

7.3.2　QAM

QAM 的含义是"M 进制正交振幅调制",它是即调相又调幅,其主要优点是频谱利用率高,也就是说可在较窄的频带内传输更多的信息。以 16QAM (16 进制正交振幅调制) 为例,理论上讲其频谱利用率可达 4 (bit/s) /Hz,而实用的 16QAM 系统的频谱利用率也可达 3 (bit/s) /Hz 左右。常用的有 16QAM、64QAM 等,而 4QAM (4 进制 QAM) 就是 QPSK。此处对 16QAM 进行仿真。

16QAM 调制的原理框图见图 7.15,其中的 2 到 4 电平变换器是进行 16QAM 调制的关

图 7.14　QPSK 解调

键所在，而其余部分与 QPSK 调制是类似的。

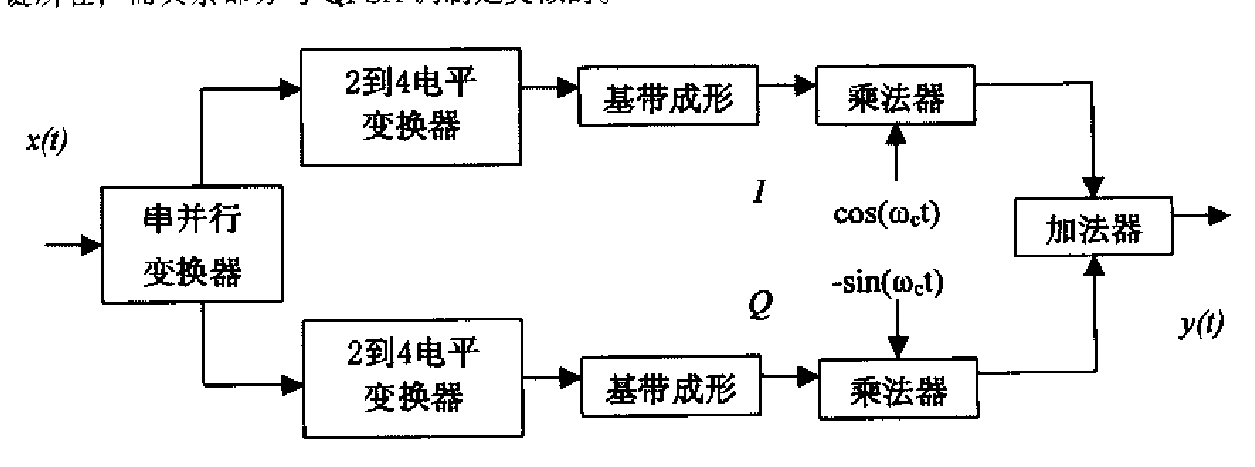


图 7.15　16QAM 调制的原理框图

表 7.2 给出了 2 到 4 电平变换的逻辑关系，而实际的信号波形如图 7.16 所示，其中采用了四个电平数值，电平级差为 1V。

表 7.2　2 到 4 电平变换的逻辑关系

	二进制码
0（-1.5V）	0　0
1（-0.5V）	0　1
2（0.5V）	1　1
3（1.5V）	1　0

16QAM 的解调是建立在 QPSK 相干解调基础之上的，只需增设两个 4 到 2 电平变换器。

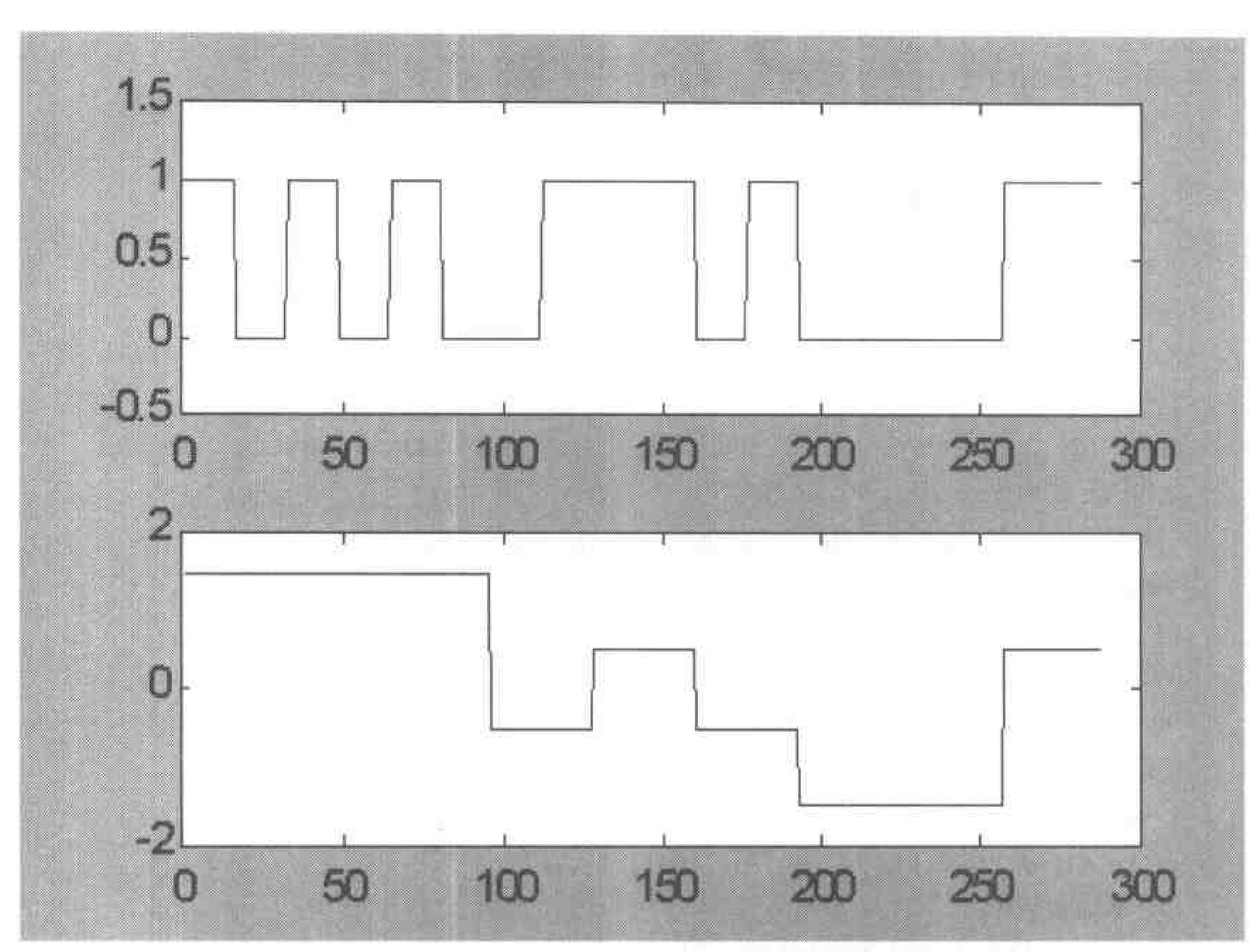


图 7.16　2 电平信号与 4 电平信号的对应关系

自定义的 M 函数 qam.m 的功能是进行 16QAM 调制，在未设定输出参数的情况下，自动显示出基带信号的波形，已调信号的波形和频谱，以及信号的星座图。该函数的用法是：

》[y, fs, fb, fc] =qam (x, Kbase, fs, fb, fc)；

x 为代表二进制码流的向量；参数 Kbase=1，表示不采用基带成形，Kbase=2，则表示采用基带成形；fs 为采样频率，fb 为比特率，fc 为载波频率，y 为已调信号。

缺省输入参数为：Kbase=2；fs=32；fb=1；fc=2，缺省输入的 x 向量确保了 16 个状态的出现。

程序清单如下：

```
function [y, fs, fb, fc] =qam (x, Kbase, fs, fb, fc);
%
%   Usage: [y, fs, fb, fc] =qam (x, Kbase, fs, fb, fc);    BBI 2000
if nargin<5; fc=2;             end;
if nargin<4; fb=1;             end;
if nargin<3; fs=32;            end;
if nargin<2; Kbase=2;          end;
if nargin<1;
  x = [1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 0];
  x= [x 1 0 1 1 1 0 1 0 1 0 0 1 1 0 0 0];
  x= [x 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0];
  x= [x 0 1 1 1 0 1 1 0 0 1 0 1 0 1 0 0];
end;
T=length (x) /fb;      m=fs/fb;     nn=length (x);
dt=1/fs;        t=0: dt: T-dt;
Kbase=rem (Kbase, 2);    Kbase=Kbase+ (Kbase= =0) *2;    tstr=' QAM';
I=x (1: 2: nn-1);             [I, In] =two2four (I, 4*m);
Q=x (2: 2: nn);               [Q, Qn] =two2four (Q, 4*m);
if Kbase= =2;
  I=bshape (I, fs, fb/4);    Q=bshape (Q, fs, fb/4);
end;
y=I.*cos (2*pi*fc*t) -Q.*sin (2*pi*fc*t); y1=y;
n=length (y);
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
if nargout<1;
  subplot (211);
  plot (t, y, t, I, t, Q, [0 T], [0 0],' b:');
  if Kbase= =2;
    tstr= [tstr ' with the RC filter'];
  end;    title (' Signal');
  n=length (y); y=fft (y) /n; y=abs (y (1: fix (n/2))) *2;
  I=find (y<1e-04); y (I) =1e-04; y=20*log10 (y);
  f1=m/n; f=0: f1: (length (y) -1) *f1;
  subplot (223);      plot (f, y,' r');     grid on;
  title (' Spectrum');    xlabel (' f / fb');     zoom xon;
  subplot (224); constel (y1, fs, fb, fc); xlabel (' Constellation');
  set (gcf,' num',' off',' name', [tstr ',' blanks (10) ' BBI 2000']);
```

```
end;
```

函数 qam.m 在运行的过程中调用了 two2four.m 函数来进行 2 电平到 4 电平的变换，该函数的程序清单如下：

```
function [y, yn] =two2four (x, m);
%
%   BBI 2000
if nargin<2; m=32; end;
if nargin<1; x= [1 0 1 0 1 0 0 1 1 1 0 1 0 0 0 0 1 1]; end;
T= [0 1; 3 2]; n=length (x); ii=1;
for i=1: 2: n-1;
  xi=x (i: i+1) +1; yn (ii) =T (xi (1), xi (2)); ii=ii+1;
end;
yn=yn-1.5;    y=yn;
for i=1: m-1;    y= [y; yn];    end;    y=y (:)';
%------------------------------------------
if nargout<1;
  subplot (211); plot (pulse (x));      v=axis; axis ( [v (1: 2) -.5 1.5]);
  subplot (212); plot (y);      v=axis; axis ( [v (1: 2) -2 2]);
end;
```

自定义的 M 函数 qamdet.m 采用相干方式进行 16QAM 解调，在未设定输出参数的情况下，自动显示出已调信号的波形、低通滤波器的输出波形、抽样判决器中经过整形的脉冲信号波形和解调出的二进制码流。该函数的用法是：

》[xn, x] =qamdet (y, fs, fb, fc);

y 为调制信号，fs 为采样频率，fb 为比特率，fc 为载波频率。xn 为解调输出的二进制码流向量，x 为解调输出的脉冲信号波形。

程序清单如下：

```
function [xn, x] =qamdet (y, fs, fb, fc);
%
%  Usage: [xn, x] =qamdet (y, fs, fb, fc);    BBI 2000
if nargin<4; fc=2;          end;
if nargin<3; fb=1;          end;
if nargin<2; fs=32;         end;
if nargin<1; y=qam ( [1 1 0 0 1 1 1 1 0 0 0 0 0 0 1 1]); end;
dt=1/fs; t=0: dt: (length (y) -1) *dt;
I=y. *cos (2*pi*fc*t);    Q=-y. *sin (2*pi*fc*t);
```

```
[b, a] =butter (2, 2*fb/fs);       tstr=' 16QAM,';
I=filtfilt (b, a, I);       Q=filtfilt (b, a, Q);
plot (i, I, i, Q);
m=4*fs/fb; N=length (y) /m; n= (.6: 1: N) *m; n=fix (n);
In=I (n);     Qn=Q (n);         xn=four2two ( [In Qn]);
nn=length (xn);                 xn= [xn (1: nn/2); xn (nn/2+1: nn)];
xn=xn (:);    xn=xn';           x= [I; Q];
%-----------------------------------------
if nargout<1;
  c=' bbbbbbbbrrrrrrrr';
  subplot (211); plot (t, y); title (' Input');
  subplot (212); plot (t, x); set (gca,' ygrid',' on'); v=axis;
  for i=1: 4*N;
    ci=rem (i, 16); ci=ci+ (ci==0) *16; ci=c (ci);
    text ( (2*i-1) *m*dt/8, v (4) *.8, int2str (xn (i)),' color', ci);
  end;   title (' Output');
  set (gcf,' num',' off',' name', …
     [' Coherent Detection of ' tstr hlanks (10) ' BBI 2000']);
  zoom xon;
end;
```

函数 qamdet.m 在运行过程中要调用 four2two.m 函数来进行 4 电平到 2 电平的变换，该函数的程序清单如下：

```
function xn=four2two (yn);
%
% BBI 2000
if nargin<1; [y, yn] =two2four; end;
y=yn; ymin=min (y); ymax=max (y); ymax=max ( [ymax abs (ymin)]);
ymin= -abs (ymax);    yn= (y-ymin) *3/ (ymax-ymin);
I0=find (yn< 0.5);                 yn (I0) =zeros (size (I0));
I1=find (yn>=0.5 & yn<1.5);        yn (I1) =ones (size (I1));
I2=find (yn>=1.5 & yn<2.5);        yn (I2) =ones (size (I2)) *2;
I3=find (yn>=2.5);                 yn (I3) =ones (size (I3)) *3;
T= [0 0; 0 1; 1 1; 1 0];           n=length (yn);
for i=1: n;
  xn (i, :) =T (yn (i) +1, :);
end;
xn=xn';    xn=xn (:);    xn=xn';
```

```
if nargout<1; disp (xn); end;
```

例：〉qam;　16QAM 调制的结果如图 7.17 所示。

〉y = qam;　qamdet (y);　16QAM 解调的结果如图 7.18 所示。

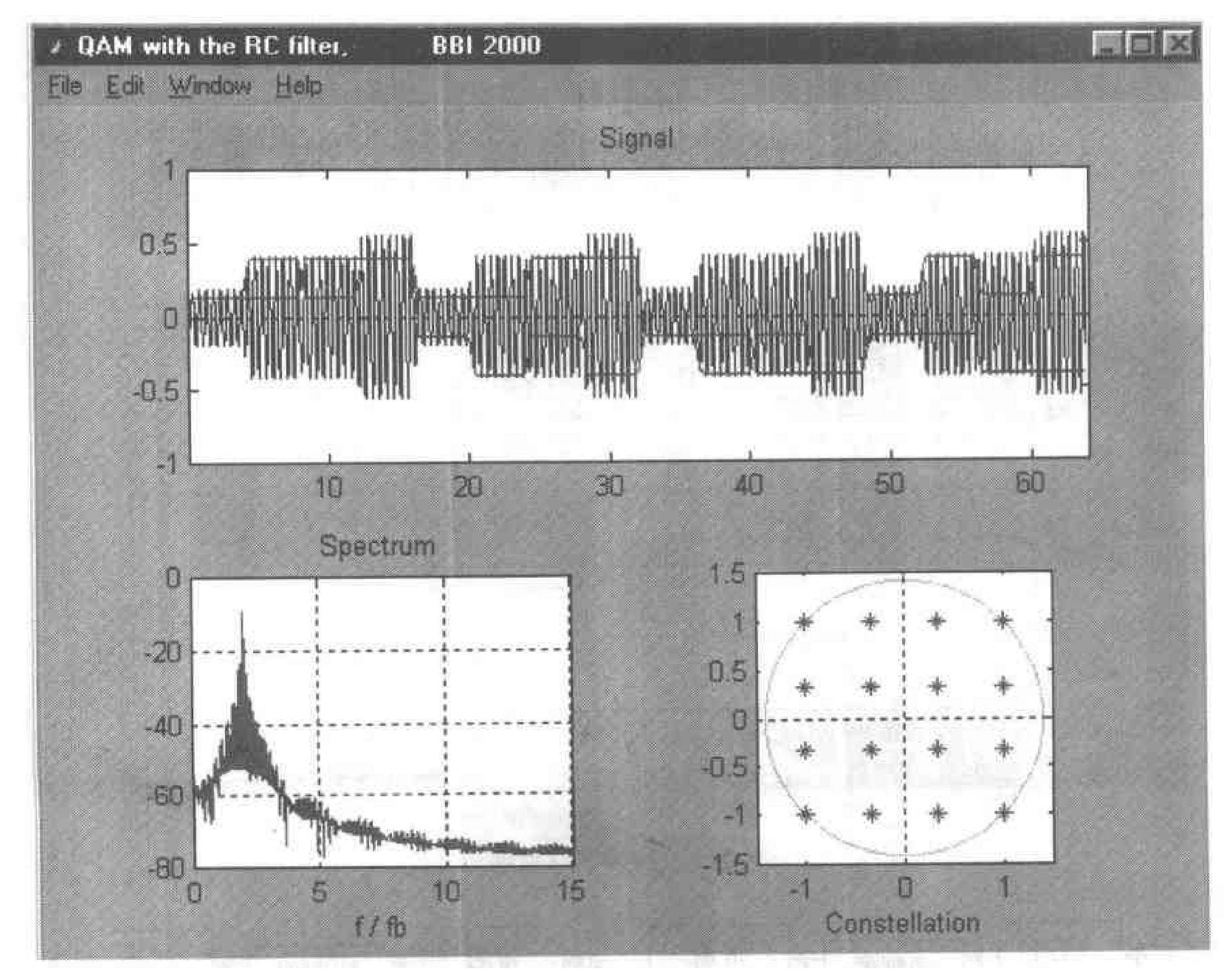


图 7.17　16QAM 调制

7.4　数字调制解调系统

数字调制与解调的系统模型如图 7.19 所示，其中信道中出现的噪声为高斯白噪声。由于数字信号在传输过程中受到噪声的干扰会产生误码，因此在数字调制系统中往往要采用信道编码器来进行差错控制编码，以减少噪声干扰的影响，常用的差错控制编码有 RS 码、卷积码等形式。

自定义的 M 函数 bmod.m 用来对数字调制系统进行仿真，其中考虑到不同的调制方式、不同的载噪比和不同的差错控制编码方式。它采用了按钮式界面，可对上述的四种二进制调制和解调的过程进行仿真，屏幕上显示出已调信号的波形和频谱、低通滤波器的输出波形、抽样判决器中经过整形的脉冲信号波形和解调出的二进制码流，并且将该码流对应的字符串显示在屏幕的下方，以观察误码产生的影响；各种参数（如滤波器带宽、载噪比、载波频

图 7.18　16QAM 解调

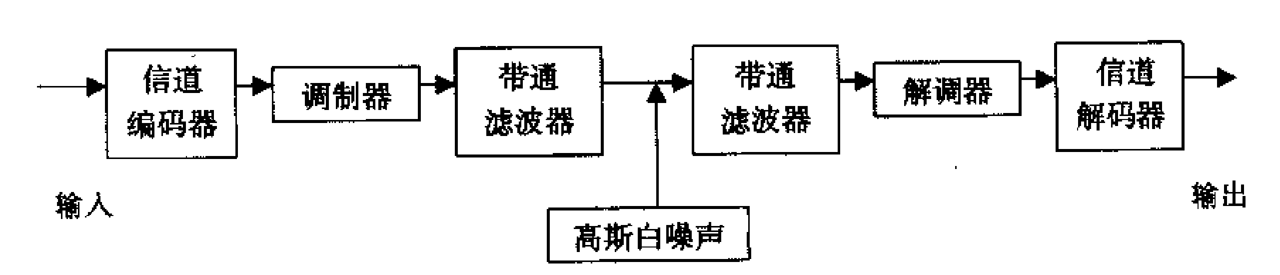


图 7.19　数字调制系统模型

率、比特率等）均可用下拉菜单和编辑框的方式进行输入，因此使用起来很方便。

各种调制方式中，带通滤波器的带宽 B 与信号比特率 f_b 的关系见表 7.3。

表 7.3　滤波器带宽与信号比特率之间的关系

	BPSK	DPSK	ASK	QPSK	QDPSK	16QAM	FSK
数字基带	$2f_b$	$2f_b$	$2f_b$	f_b	f_b	$0.5f_b$	$f_{c2}-f_{c1}+2f_b$
基带成形（$\alpha=0.5$）	$1.5f_b$	$1.5f_b$	$1.5f_b$	$0.75f_b$	$0.75f_b$	$0.375f_b$	$f_{c2}-f_{c1}+1.5f_b$

频谱利用率可以从带通滤波器带宽和信号比特率求出，

$$\eta = \frac{f_b}{B} \qquad \text{(bit/s)/Hz}$$

M 函数 bmod.m 支持三种差错控制编码方式，即 RS 码、汉明码、BCH 码。在 COMM 工具箱内设有 encode.m 和 decode.m 两个 M 函数用来进行差错控制编码和解码，在调用这两个函数时要输入相应的码长 m 和信息位数 k。

M 函数 bomd.m 的使用是很简单的，在 MATLAB 命令窗口内只要键入 bmod；便可运行。该函数的程序清单如下：

```
function [] =bmod (action);
%
% BBI 2000
global x y Kmod Kbase fs fb fc fc1 fc2 rd Kmd B w CNR Kch msg b a…
      H422 T eta yr yn y0 k H1 H74 H75 H92 Kerror Kecc …
      H101 H112 H122 NN KK
if nargin<1; action=' initialized'; end;
if strcmp (action,' initialized');
   figure (' Num',' off',' Units',' pix',' pos', [5 29 792 530], …
           ' Color', [1 1 1]);
   uicontrol (' Style',' Frame',' Units',' Normal',' Position', …
              [.82 0 .2 1],' Back', [.8 .8 .8]);
   H1=uicontrol (' Style',' Popup',' Units',' Normal',' Position', …
                 [.84 .93 .15 .05],' String', str2mat (' BPSK',' DPSK', …
                 ' FSK',' ASK',' QPSK',' QDPSK',' 16QAM'), …
                 ' Back', [1 1 1],' Callback',' bmod method;');
   H2=uicontrol (' Style',' Popup',' Units',' Normal',' Position', …
                 [.84 .86 .15 .05],' String', str2mat (' Baseband Shaping', …
                 ' Baseband'),' Back', [1 1 1],' Callback',' bmod base;');
  H3=uicontrol (' Style',' Popup',' Units',' Normal',' Position', …
             [.84 .79 .15 .05],' String', str2mat (' Ref digit=0', …
             ' Ref digit=1'),' Back', [1 1 1],' Call',' bmod refd;');
  H411=uicontrol (' Style',' Text',' Units',' Normal',' Position', …
                 [.822 .725 .09 .04], …
                 ' String',' B (MHz) :',' Back', [.8 .8 .8]);
  H422=uicontrol (' Style',' Edit',' Units',' Normal',' Position', …
                 [.91 .73 .08 .04],' Back', [1 1 1], …
                 ' String',' 1.5'' Call',' bmod bandwidth;');
  H41=uicontrol (' Style',' Text',' Units',' Normal', …
                 ' Position', [.822 .645 .09 .04], …
```

```
                ' String',' Rb (Mb/s) :',' Back', [.8 .8 .8]);
H42 = uicontrol (' Style',' Edit',' Units',' Normal', ...
                ' Position', [.91 .65 .08 .04], ...
                ' Back', [1 1 1],' String',' 1',' Call',' bmod fd;');
H51 = uicontrol (' Style',' Text',' Units',' Normal', ...
                ' Position', [.822 .575 .09 .04], ...
                ' String',' fc (MHz):',' Back', [.8 .8 .8]);
H52 = uicontrol (' Style',' Edit',' Units',' Normal', ...
                ' Position', [.91 .58 .08 .04], ...
                ' Back', [1 1 1],' String',' 2',' Call',' bmod fc;');
H61 = uicontrol (' Style',' Text',' Units',' Normal', ...
                ' Position', [.822 .515 .09 .04], ...
                ' String',' fc2 (MHz):',' Back', [.8 .8 .8]);
H62 = uicontrol (' Style',' Edit',' Units',' Normal', ...
                ' Position', [.91 .52 .08 .04], ...
                ' Back', [1 1 1],' String',' 2.5',' Call',' bmod fc2;');
H71 = uicontrol (' Style',' Text',' Units',' Normal', ...
                ' Position', [.822 .445 .09 .04], ...
                ' String',' fs (MHz):',' Back', [.8 .8 .8]);
H72 = uicontrol (' Style',' Edit',' Units',' Normal', ...
                ' Position', [.91 .45 .08 .04], ...
                ' Back', [1 1 1],' String',' 32',' Call',' bmod fs;');
H75 = uicontrol (' Style',' Checkbox',' Units',' Normal', ...
                ' Pos', [.825 .4 .168 .04], ...
                ' String',' Communication Channel',' value', 1, ...
                ' Back', [.8 .8 .8],' Call',' bmod channel;');
H73 = uicontrol (' Style',' Text',' Units',' Normal', ...
                ' Position', [.822 .345 .09 .04], ...
                ' String',' C/N (dB):',' Back', [.8 .8 .8]);
H74 = uicontrol (' Style',' Edit',' Units',' Normal', ...
                ' Position', [.91 .35 .08 .04], ...
                ' Back', [1 1 1],' String',' 5',' Call',' bmod CNR;');
H101 = uicontrol (' Style',' Popup',' Units',' Normal',
                ' Position', [.825 .28 .18 .05], ...
                ' String', str2mat (' No Eerror Control Coding', ...
                ' Reed - Solomon',' Hamming',' BCH'), ...
                ' Back', [1 1 1],' Callback',' bmod ECC;');
H111 = uicontrol (' Style',' Text',' Units',' Normal', ...
                ' Position', [.822 .225 .09 .04], ...
```

```
                  'String','n=2^m-1:','Back',[.8 .8 .8]);
  H112=uicontrol('Style','Edit','Units','Normal',…
                  'Position',[.91 .23 .08 .04],…
                  'Back',[1 1 1],'String','63','Call','bmod nn;');
H121=uicontrol('Style','Text','Units','Normal',…
                  'Position',[.822 .165 .09 .04],…
                  'String','k:','Back',[.8 .8 .8]);
H122=uicontrol('Style','Edit','Units','Normal',…
                  'Position',[.91 .17 .08 .04],…
                  'Back',[1 1 1],'String','47','Call','bmod kk;');
H8=uicontrol('Style','Push','Units','Normal',…
                  'Position',[.84 .11 .15 .04],…
                  'String','MOD/DEMOD','Call','bmod demod;');
H91=uicontrol('Style','Text','Units','Normal',…
                  'Position',[.84 .06 .15 .04],…
                  'String','Message','Back',[.8 .8 .8]);
H92=uicontrol('Style','Edit','Units','Normal',…
                  'Position',[.822 .03 .178 .04],…
          'Back',[1 1 .2],'String','CHN','Call','bmod message;');
%-----------------------------------------
Kmod=1; Kbase=2; fs=32; fb=1; fc=2; rd=0; fc2=2.5;
Kmd=1; B=1.5*fb; CNR=5; Kch=1; msg='CHN'; eta=fb/B;
T=[2 2 2 2 1 1 .5; 1.5 1.5 1.5 1.5 .75 .75 .375];
w=[fc-.5*B fc+.5*B]/(fs/2); Kerror=0; Kecc=1; NN=63; KK=47;
elseif strcmp(action,'method');
  Kmod=get(gco,'value'); Kmd=1;
  if Kmod==3;
          w=(fc+fc2)/2; B=(fc2-fc)+T(Kbase)*fb;
          w=[w-.5*B w+.5*B]/(fs/2);
else;
          B=T(Kbase,Kmod)*fb;      w=[fc-.5*B fc+.5*B]/(fs/2);
end;      set(H422,'string',num2str(B));
elseif strcmp(action,'base');
  Kbase=get(gco,'value'); Kbase=3-Kbase; Kmd=1;
  if Kmod==3;
          w=(fc+fc2)/2; B=(fc2-fc)+T(Kbase)*fb;
          w=[w-.5*B w+.5*B]/(fs/2);
else;
  B=T(Kbase,Kmod)*fb;      w=[fc-.5*B fc+.5*B]/(fs/2);
```

```
    end;   set (H422,' string', num2str (B));
  elseif strcmp (action,' refd');
    K=get (gco,' value'); rd=K-1; Kmd=1;
  elseif strcmp (action,' bandwidth');
    B=get (gco,' string');     B=str2num (B); Kmd=1;
    if Kmod= =3;               w= [w-.5*B w+.5*B] / (fs/2);
    else;                      w= [fc-.5*B fc+.5*B] / (fs/2);
    end;
  elseif strcmp (action,' fb');
    fb=get (gco,' string'); fb=str2num (fb); Kmd=1;
  elseif strcmp (action,' fc');
    fc=get (gco,' string'); fc=str2num (fc); Kmd=1;
  elseif strcmp (action,' fc2');
    fc2=get (gco,' string'); fc2=str2num (fc2); Kmd=1;
  elseif strcmp (action,' fs');
    fs=get (gco,' string'); fs=str2num (fs); Kmd=1;
  elseif strcmp (action,' message');
    msg=get (gco,' string'); x=str2cod (msg); Kmd=1;
  elseif strcmp (action,' demod');
    Kmd= -Kmd;
  elseif strcmp (action,' CNR');
    K=get (gco,' string'); CNR=str2num (K); if Kch= =1; Kmd=1; end;
  elseif strcmp (action,' channel');
    Kch=get (gco,' value'); Kmd=1;
  elseif strcmp (action,' ECC');
    Kecc=get (gco,' value');    Kmd=1;
    if Kecc= =4; KK=45; set (H122,' string', num2str (KK)); end;
  elseif strcmp (action,' nn');
    NN=get (gco,' string'); NN=str2num (NN); Kmd=1;
  elseif strcmp (action,' kk');
    KK=get (gco,' string'); KK=str2num (KK); Kmd=1;
    if Kecc= =1 | Kecc= =3;
    Kecc=2;    set (H101,' value', 2);
    end;
  end;
  x=str2cod (msg); n1=0;
  %-----------------------------------------------------
  if Kecc= =2;                                  %   Error Control Coding
        x=encode (x, NN, KK,' rs')';
```

```
    elseif Kecc==3;
        m=log (NN+1) /log (2);              KK=NN-m;
        x=encode (x, NN, KK,' hamming')';       set (H122,' string', num2str
(KK));
    elseif Kecc==4;
        x=encode (x, NN, KK,' bch')' ;
    end; n1=rem (length (x), 8);
    if n1>0;    x= [x zeros (1, 8-n1)]; end;
    %------------------------------------------------
    if Kmd==1 & Kerror==0;                      % Modulate
      if Kmod<=2;
          y=bpsk (x, Kmod, Kbase, fs, fb, fc, rd);
        elseif Kmod==3;
          y=fsk (x, Kbase, fs, fb, fc, fc2);
    elseif Kmod==4;
      y=ask (x, Kbase, fs, fb, fc);
    elseif Kmod==5 | Kmod==6;
      y=qpsk (x, Kmod, Kbase, fs, fb, fc);
    elseif Kmod==7;
      y=qam (x, Kbase, fs, fb, fc);
    end;   y0=y;   yr=y; eta=fb/B;    N=length (y); k=1;
    if w (1) <eps; w (1) =eps; end;      if w (2) >1; w (2) =1; end;
    if Kch==1;
       [b, a] =cheby1 (2, .1, w); y=filtfilt (b, a, y); % Tx Bandpass Filter    yn=
randn (1, N);        yn1=filtfilt (b, a, yn);
      k=sqrt (1e-07/ (sum (yn1.*yn1) /N));
      yn=k*yn;         Pn=k*k*sum (yn1.*yn1);
      y1=filtfilt (b, a, y);         Pc=sum (y1.*y1);
      k=sqrt (Pn/Pc*10^ (CNR/10)); y=k*y; y=y+yn;
      yr=filtfilt (b, a, y);               % Rx Bandpass Filter
    end;
      bmod1 (x, y/k, Kmod, Kbase, fs, fb, fc, rd);
      bmod2 (x, y, Kmod, fs, fb, fc, k*y0);
    elseif Kmd==-1;                     % Demodulate
      if Kmod<=2;                       [xn, x1] =bpskdet (yr, Kmod, fs, fb, fc, rd);
      elseif Kmod==3;                   [xn, x1] =fskdet (yr, fs, fb, fc, fc2);
      elseif Kmod==4;                   [xn, x1] =askdet (yr, fs, fb, fc);
      elseif Kmod==5 |Kmod==6;          [xn, x1] =qpskdet (yr, Kmod, fs, fb, fc);
      elseif Kmod==7;                   [xn, x1] =qamdet (yr, fs, fb, fc);
```

```
end;
% --------------------------------------------
m2=length (x); I=find (x=xn); m1=length (I);
if n1>0; xn=xn (1: length (xn) -8+n1); end;
if Kecc= =2;                    %Error Control Coding
  xn=decode (xn, NN, KK,' rs')';
elseif Kecc= =3;
  xn=decode (xn, NN, KK,' hamming')';
elseif Kecc= =4;
  xn=decode (xn, NN, KK,' bch')';
end;
hmod3 (xn, x1, Kmod, fs, fb, msg, [m1 m2]);
end;
```

bmod.m 函数在运行过程中要调用 bmod1.m，bmod2.m 和 bmod3.m 函数，它们的功能主要是绘制图形。

hmod1.m 的程序清单如下：

```
function [] =bmod1 (x, y, Kmod, Kbase, fs, fb, fc, rd);
%
%       BBI 2000
global eta
mstr=str2mat (' BPSK',' DPSK',' FSK',' ASK (OOK)', ...
              ' QPSK',' QDPSK',' 16QAM');
tstr=mstr (Kmod, :);
if Kmod= =2;
  x=dcode (x, rd);
  tstr= [tstr ' (reference digit: ' int2str (rd) ')'];
end;
eta=round (eta*1000) *.001;
set (gcf,' name', [tstr ', Spectral Efficiency: ' ...
    num2str (eta) ' (bit/s) /Hz' blanks (10) ' BBI 2000']);
n=length (y); t= (0: n-1) /fs;
x=pulse (x, fs/fb, 2);
if Kbase= =2;    x=bshape (x);    end;
subplot (' position', [.05 .5811 .74 .3439]);
plot (t, y, t, x*.5);    tstr=' Signal';
if Kbase= =2; tstr= [tstr ' ( alfa = 0.5 )']; end;
title (tstr); xlabel (' t (μs)'); zoom xon;
```

bmod2.m 和程序清单如下：

```
function [] =bmod2 (x, y, Kmod, fs, fb, fc, y1);
%
%    BBI 2000
n=length (y);
yy=fft (y) /n;    yy=abs (yy (1: fix (n/2))) *2;
I=find (yy<4e-7);     yy (I) =4e-7;     yy=20*log10 (yy) +10;
yy1=fft (y1) /n;    yy1=abs (yy1 (1: fix (n/2))) *2;
I=find (yy1<4e-7); yy1 (I) =4e-7;    yy1=20*log10 (yy1) +10;
f1=fs/n;    f=0: f1: (length (yy) -1) *f1;
if Kmod= =3 | Kmod= =4;
  subplot (' position', [0.05 0.11 0.74 0.3439]);
  plot (f, yy1, f, yy,' r');    grid on;
  h=get (gca,' children');    set (h (2),' color', [0 .99 .99]);
  title (' Spectrum (dBm)');    xlabel (' f (MHz)');    zoom xon;
elseif Kmod<3;
  suhplot (' position', [0.05 0.11 0.45 0.3439]);
  plot (f, yy1, f, yy,' r');    grid on;
  h=get (gca,' cbildren');    set (h (2),' color', [0 .99 .99]);
  title (' Spectrum (dBm)');    xlabel (' f (MHz)'); zoom xon;
  ph=phdet (y1, fs/fc);
  subplot (' position', [0.52 0.11 0.3 0.3439]);
  H=polar (ph, ones (size (ph)),' ro');
  set (H,' markers', 8,' linew', 2);    xlabel (' Phase');
elseif Kmod>4;
  subplot (' position', [0.05 0.11 0.45 0.3439]);
  plot (f, yy1, f, yy,' r');    grid on;
  h=get (gca,' children');    set (h (2),' color', [0 .99 .99]);
  title (' Spectrum (dBm)');    xlabel (' f (MHz)'); zoom xon;
  subplot (' position', [0.5 0.11 0.344 0.344]);
  constel (y1, fs, fb, fc);    xlabel (' Constellation');
end; zoom xon;
```

bmod3.m 的程序清单如下：

```
function Kerror=bmod3 (xn, x, Kmod, fs, fb, msg, nb);
%
```

```
% Usege: bmod3 (xn, x, Kmod, fs, fb, msg); BBI 2000
m=fs/fb; n=length (x (1,:)); t= (0: n-1) /fs; [mm, nn] =size (x);
xm=max (x); xm=max (xm (:)); x=.51*x/xm;
subplot (' position', [0.05 0.11 0.74 0.3439]);
if mm==1;
  plot (t, x, t, sign (x) *.5);
elseif mm==2;
  xm=max (abs (x (1: 2, 1: fix (nn*.95)))); x=.6*x/max (xm); plot (t, x);
end;
v=axis; axis ( [v (1: 2) -1 1]); set (gca,' ygrid',' on');
if Kmod==7;
  set (gca,' ytick', [-1 -.6 -.2 0.2 .6 1]);
end;
c=' bbbbbbbbrrrrrrrr'; dt=t (2);
for i=1: length (xn);
  ci=rem (i, 16); ci=ci+ (ci==0) *16; ci=c (ci);
  text ( (2*i-1) *m*dt/2, .8, int2str (xn (i)),' color', ci);
end;
title (' Detection Output');
N=length (str2cod (msg));
if nb (1) >0;
  bstr= [' BER: ' num2str (nb (1)) ' / ' num2str (nb (2)) ' '];
  text (v (2), 1.15, bstr,' color',' r',' fonts', 10,' hor',' right');
end;
mstr=cod2str (xn (1: N)); K=strcmp (mstr, msg) +1; c=' rb';
text (0, -1.4,' Demod Message:');
text (v (2) *3/16, -1.4, mstr,' color', c (K),' fontsize', 10);
Kerror=rem (K, 2);
```

使用 bmod.m 函数可以对数字调制解调系统进行仿真，从仿真的结果中可以清楚地看出噪声对误码产生的影响。不同的调制方式的频谱利用率是不同的，同时不同的调制方式对接收端载噪比的要求也是不一样的，表 7.4 给出了在未采用差错控制编码的情况下通过多次仿真而归纳出的结果，此结果与一般理论分析的结论是吻合的。

表 7.4 调制方式与系统载噪比（未设信道编码器）

	BPSK	DPSK	FSK	QPSK	16QAM
C / N	6 dB	6 dB	9 dB	9 dB	23 dB

在噪声过大的情况下，解调时就会产生误码，为了醒目，此时解码后得到的字符串用红

色表示，而在无误码的情况下，字符串用蓝色表示。

图 7.20 为 QPSK 调制的结果，采用了 $\alpha=0.5$ 的基带成形滤波器，比特率为 1Mbit/s，载波频率为 2MHz，输出滤波器的带宽为 0.75MHz，因此频谱利用率为 1.333（bit/s）/Hz，信道的载噪比为 5.5dB，传送的信息为 ' MATLAB 3.5，4.2，5.2 and 5.3. £30 ¥451μV straβ [mæp] metà è così casinò più '。

图 7.21 为 QPSK 解调的仿真结果。在载噪比为 5.5dB 情况下，误码为 15 个，而码共有 1136 个，由于采用了 RS（63，47）码作为差错控制编码，故有效地对误码进行了纠错。

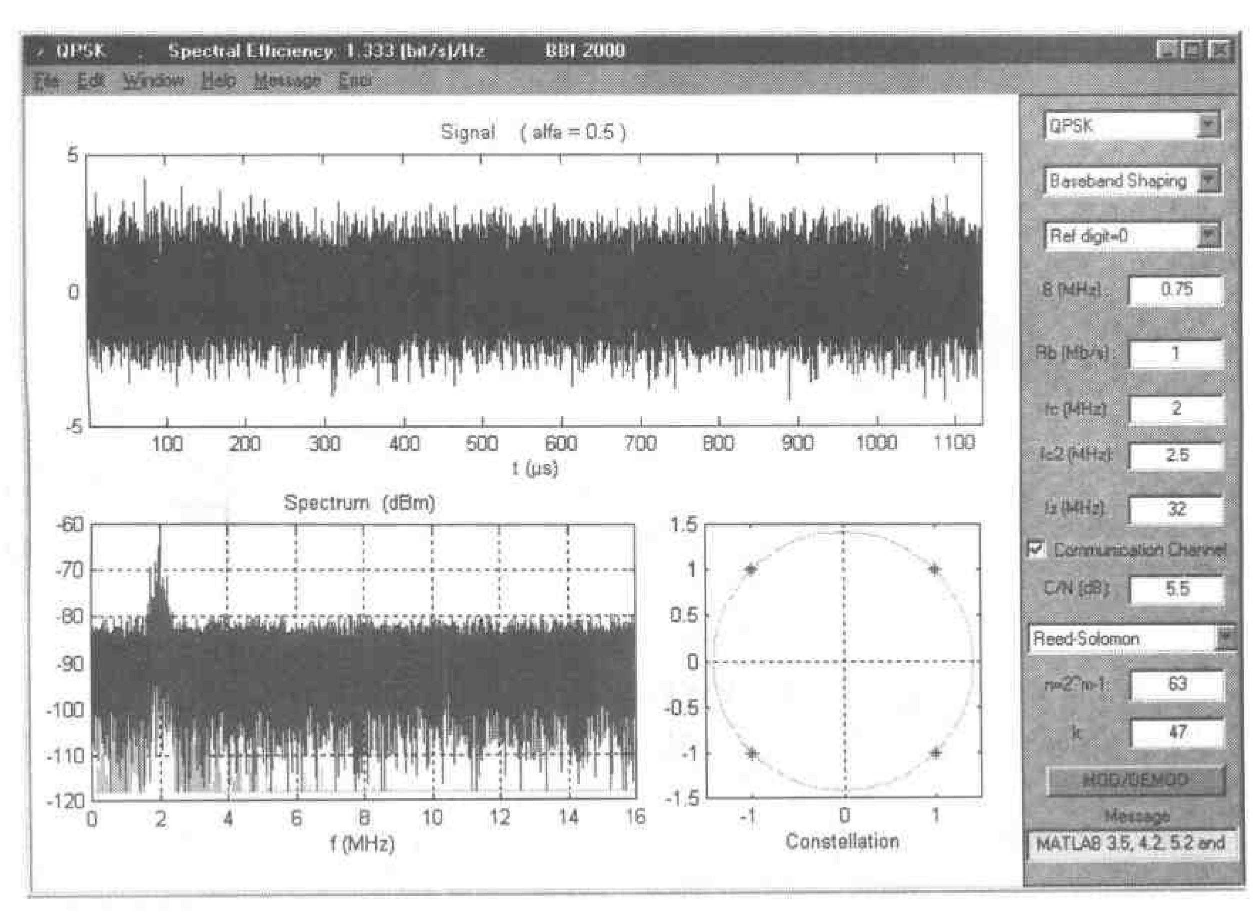


图 7.20　数字调制系统

7.5　数字卫星电视接收系统

衡量一个数字卫星电视接收系统质量的主要参数是 E_b/N_0，其中 E_b 为每个比特的能量，而 N_0 代表噪声功率谱密度。E_b/N_0的计算公式为

$$E_b/N_0=\begin{cases}EIRP-L_s-\Delta L-\beta+G_{ANT}-10\lg(T_A+T_{LNB})-10\lg R_b-10\lg k & (MCPC)\\ EIRP-BO-10\lg m-L_s-\Delta L-\beta+G_{ANT}-10\lg(T_A+T_{LNB})-10\lg R_b-10\lg k & (SCPC)\end{cases}$$

其中，EIRP 为卫星转发器的等效全向辐射功率，单位为 dBW；L_S 为电波在自由空间内的传播损耗；ΔL 为卫星链路的附加损耗；β 为上行链路因子；G_{ANT}为接收天线的增益；T_A 为接收天线的等效噪声温度；T_{LNB}为高频头的噪声温度；R_b 为信号的比特率，k 为玻尔兹曼常数（$k=1.38\times10^{-23}$）。

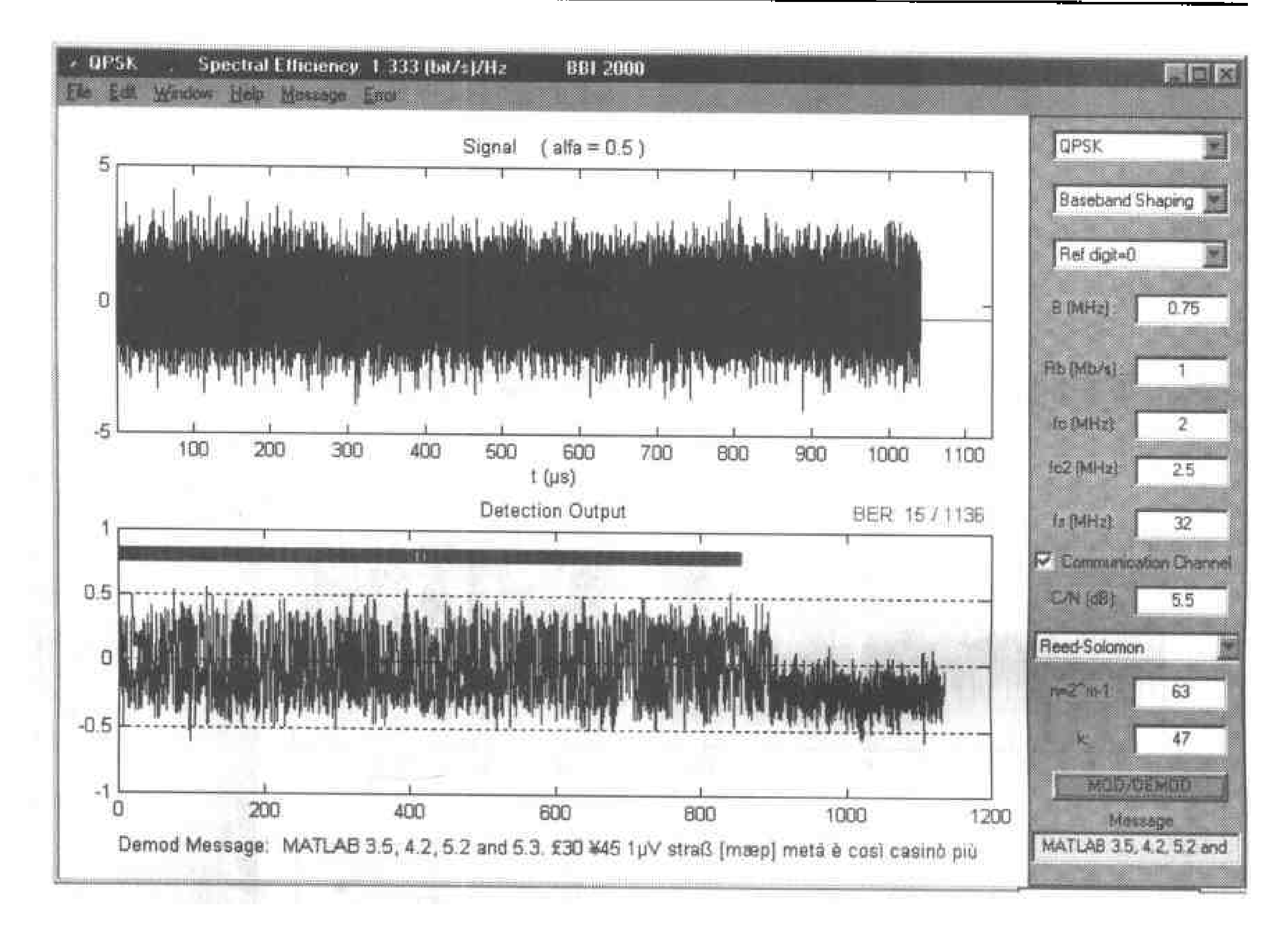


图 7.21 数字解调系统

在数字卫星电视广播中，具体有多路单载波（MCPC）和单路单载波（SCPC）两种工作方式，当采用 SCPC 方式时，必须考虑功率回退因子 BO，以及每个转发器转发的载波数 m。

当 E_b/N_0 的数值大于门限以后，信号的质量就得到了保证。DVB－S 系统中规定的门限值见表 7.5。

表 7.5 DVB－S 系统误码性能要求

内码编码率	1/2	2/3	3/4	5/6	7/8
E_b/N_0门限	4.5 dB	5.0 dB	5.5 dB	6.0 dB	6.4 dB

自定义的 M 函数 satdtv.m 用来对数字卫星电视链路进行计算，同时根据接收地点的位置和所接收的卫星位置计算出接收天线的仰角 EL、方位角 AZ 和极化角。

satdtv.m 函数的用法是：

```
》[EbN0, EL, AZ, alfa, R, G, GT, Pc] = satdtv (lat, lnt, EIRP, lnts, band, d, …
                                              e1, Tr, mod, Rs, FEC, BO, m, dL);
```

输入参数：

lat　接收地点的纬度（°），　　lnt　接收地点的经度（°）

EIRP　卫星的等效全向辐射功率（dBW），　　lnts　卫星的经度（°）

band 波段（字符串 ' C' 或 ' Ku' ），
d 卫星接收天线的直径（ m ）
e1 卫星接收天线的口面效率，
Tr 高频头的噪声温度（ K ）
mod 工作方式（字符串' M' 或 ' S'），
R_s 符号率（MS/s）
FEC 内码编码率，
BO 功率回退（dB）
m 转发器转发的载波数，
dL 卫星链路的附加损耗（ dB ）

缺省的输入参数为：

satdtv(39.9,116.5,39,100.5,'C',1.5,0.6,30,'S',4.42,3/4,4,5,.5)；

输出参数：

EbN0 单位为（ dB ），
EL 接收天线的仰角（°）
AZ 接收天线的方位角（°），
alfa 接收天线的极化角（°）
R 接收点与卫星之间的距离（ km ），
G 卫星接收天线的增益（ dB ）
GT 卫星接收系统的优值（dB），
Pc 接收天线输入功率（ dBm ）

sattv.m 函数的程序清单如下：

```
function [EbN0, EL, AZ, alfa, R, G, GT, Pc] =satdtv (lat, lnt, EIRP, lnts, …
                              band, d, e1, Tr, mod, Rs, FEC, BO, m, dL);
%   BBI 2000
% Usage: [EbN0, EL, AZ, alfa, R, G, GT, Pc] =satdtv (lat, lnt, EIRP, …
                         lnts, band, d, e1, Tr, Kmod, Rs, FEC, BO, m, dL);
% Default: [EbN0, EL, AZ, alfa, R, G, GT, Pc] =satdtv (39.9, 116.5, 39, …
                         100.5,' C', 1.5, 0.6, 30,' S', 4.42, 3/4, 4, 5, .5);
if nargin<1; lat=39.9;          end;
if nargin<2; lnt=116.5;         end;
if nargin<3; EIRP=39;           end;
if nargin<4; lnts=100.5;        end;
if nargin<5; band=' C';         end;
if nargin<6; d=1.5;             end;
if nargin<7; e1=.6;             end;
if nargin<8; Tr=30;             end;
if nargin<9; mod=' S';          end;    %    SCPC
if nargin<10; Rs=4.42;          end;    %    in MS/s
if nargin<11, FEC=3/4;          end;
if nargin<12; BO=4;             end;
if nargin<13; m=5;              end;
if nargin<14; dL=0.5;           end;
a=6.6108;
ph= (lnt-lnts) * pi/180;               costh=cos (ph) * cos (lat * pi/180);
AZ-tan (ph) /sin (lat * pi/180);       AZ=atan (AZ) * 180/pi;
EL= (costh-1/a) /sqrt (1-costh^2);     EL=atan (EL) * 180/pi;
```

```
alfa = sin (ph) /tan (lat * pi/180);          alfa = atan (abs (alfa)) * 180/pi;
R = 6378 * sqrt (a * a + 1 - 2 * a * costh);
if strcmp (band, ' C');
  lmd = .075;     Ta = 16 + 100 * EL.^ ( - .75);
elseif strcmp (band, ' Ku');
  lmd = .025;        Ta = 24 + 250 * EL.^ ( - 1);
end;
Rb = Rs * 2 * FEC * 1e + 06;                 % bit rate
G = 20 * log10 (pi * d/lmd) + 10 * log10 (el);
Ls = 20 * log10 (4 * pi * R * 1000/lmd);
GT = G - 10 * log10 (Ta + Tr);     beta = .5;
EbN0 = EIRP - Ls - abs (dL) - beta + GT - 10 * log10 (Rb) + 228.6;
if strcmp (mod, ' s') |strcmp (mod, ' S');
  EbN0 = EbN0 - BO - 10 * log10 (m);
end;
Pc = EIRP - Ls - abs (dL) - beta + G + 30;     % dBm
```

例　在北京接收亚洲二号卫星 C 波段转发器转发的江苏卫视节目，该节目为 SCPC 方式，符号率为 4.42MS/s，内码编码率为 3/4。

北京地区的经度为 116.5°E，纬度为 39.9°N；亚洲二号卫星的位置为 100.5°E，等效全向辐射功率为 39dBW，每个转发器转发 5 套单路单载波节目，功率回退为 4dB。采用直径为 1.5m 的接收天线，设天线的口面效率为 50%，并选用 30K 的高频头。将上述参数代入后，可得计算结果如下：

```
》[EbN0, EL, AZ, alfa, R, G, GT, Pc] = satdtv (39.9, 116.5, 39, 100.5, …
'C', 1.5, 0.5, 30,'S', 4.42, 3/4, 4, 5);

EbN0 =
  7.1622e + 000
EL =
  4.0954e + 001
AZ =
  2.4086e + 001
alfa =
  1.8245e + 001
R =
  3.7707e + 004
G =
  3.2953e + 001
GT =
```

1.5779e+001

Pc =

-9.5058e+001

从计算结果可以看到采用 1.5m 的接收天线时，Eb/N0 尚有 1.6dB 的余量。

参考文献

[1] 樊昌信，詹道庸，徐炳祥，吴成柯．通信原理（第四版）．北京：国防工业出版社，1995

[2]（美）Leon W Couch. Digital and Analog Communication Systems. 北京：清华大学出版社，1998

[3]（美）帕普里斯 A. 信号分析．北京：科学出版社，1981

[4]（美）费赫 K. 数字微波通信系统．北京：国防工业出版社，1988

[5] 车晴，王京玲．数字卫星广播系统．北京：北京广播学院出版社，2000